



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VÝVOJ TESTOVACÍHO MODELU MANET SÍTĚ V PROSTŘEDÍ NS-3

DEVELOPMENT OF EVALUATION MODEL OF MANET IN NS-3 ENVIRONMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ALEŠ GILÍK

VEDOUcí PRÁCE

SUPERVISOR

Ing. JIŘÍ HOŠEK, Ph.D.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Aleš Gilík

ID: 119411

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Vývoj testovacího modelu MANET sítě v prostředí NS-3

POKYNY PRO VYPRACOVÁNÍ:

V rámci bakalářské práce nejprve nastudujte problematiku mobilních ad hoc sítí (MANET). Zaměřte se zejména na princip komunikace v těchto sítích a nejpoužívanější směrovací protokoly. V rámci praktické části se seznámte se simulačním prostředím NS-3 a vytvořte v něm základní model MANET sítě tvořený minimálně 10 mobilními uzly. Vývoj simulačního modelu koncipujte tak, že vytvoříte centrální konfigurační soubor, ve kterém se provede konfigurace simulace a následně se provede simulace s požadovanými parametry. Pro práci s centrálním konfiguračním souborem navrhnete a vytvoříte webové uživatelské rozhraní.

DOPORUČENÁ LITERATURA:

[1] ILYAS, M.: The Handbook of Ad Hoc Wireless Networks. Boca Raton: CRC Press, 2003, ISBN: 0-8493-1332-5.

[2] MOHAPATRA, P., KRISHNAMURTHI, S.: Ad Hoc Networks: Technology and Protocols. Boston: Springer Press, 2005, ISBN: 0-387-22689-3.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Jiří Hošek, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ANOTACE

Bakalářská práce se zaměřuje na sítě Manet a vytvoření webové aplikace pro generování konfiguračního souboru, usnadňujícího simulování Manet sítí. V teoretické části obsahuje popis Manet sítí, dělení směrovacích protokolů a popis pěti vybraných z nich. V práci je taktéž popsán simulátor Ns-3, jeho používané abstrakce a práce se simulacemi. Praktická část popisuje vytvoření konfiguračního souboru a jeho součásti. Pro vytváření konfiguračního souboru byla zhotovena webová aplikace v jazyku PHP. Všechny použité technologie pro vytvoření webové aplikace jsou v práci popsány. Webová aplikace je přehledně zpracována a po zadání parametrů simulace a Manet sítě nabídne uživateli ke stažení vygenerovaný konfigurační soubor.

KLÍČOVÁ SLOVA

Manet, Ad hoc, Ns-3, směrovací protokoly, konfigurační soubor, PHP, Nette framework, webová aplikace, generátor

ABSTRACT

This bachelor thesis is focused on Manet networks and creation of web application for generate configuration file which is simplifying simulation of Manet networks. The theoretical part concerns with description of Manet networks, distribution of routing protocols and closer look on five of the protocols. In the next part is described simulator Ns-3 with his content for simulations. The practical part is describing creation of configuration file and his elements. For that was created web application in PHP language. All used technologies for creation of the application are described in this thesis. The web application is fully processed and operational.

KEYWORDS

Manet, Ad hoc, Ns-3, routing protocols, configuration file, PHP, Nette framework, web application, generator

GILÍK, A. *Vývoj testovacího modelu MANET sítě v prostředí NS-3*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 57 s. Vedoucí bakalářské práce Ing. Jiří Hošek, Ph.D..

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Vývoj testovacího modelu MANET sítě v prostředí NS-3“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 3.6.2013

.....
podpis autora

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Poděkování

Děkuji svému vedoucímu bakalářské práce Ing. Jiřímu Hoškovi, Ph.D. za odborné vedení, dobrý přístup ke studentům a výborný motivační systém při psaní mé bakalářské práce. Dále bych chtěl poděkovat Ing. Pavlu Vajsarovi taktéž za odborné vedení, rady a kontrolu mé bakalářské práce.

V Brně dne 3.6.2013

.....

podpis autora

OBSAH

Seznam tabulek	11
Úvod	12
1 Manet síť	13
1.1 Co jsou a jak vznikly	13
1.2 Mobile ad hoc síť	13
1.3 Dělení protokolů pro Manet síť	14
1.3.1 Proaktivní směrovací protokoly	15
1.3.2 Reaktivní směrovací protokoly	15
1.3.3 Hybridní směrovací protokoly	15
1.4 Nejvýznamnější směrovací protokoly pro MANET síť	16
1.4.1 Protokol OLSR	16
1.4.2 Protokol DSDV	16
1.4.3 Protokol DSR	17
1.4.4 Protokol AODV	17
1.4.5 Protokol ZRP	17
2 Simulátor Ns-3	18
2.1 Co je to simulátor Ns-3	18
2.2 Výhody oproti jiným simulátorům	18
2.3 Spouštění simulací v Ns-3	19
2.4 Používané abstrakce	19
2.4.1 Uzel	20
2.4.2 Aplikace	20
2.4.3 Kanál	20
2.4.4 Síťové zařízení	20
2.4.5 Pomocníci topologie (helpers)	20
2.5 Definování mobility v Ns-3	20
2.6 Modely k nastavení mobility uzlů	20
2.6.1 GaussMarkovMobilityModel	21

2.6.2	RandomDirection2dMobilityModel	22
2.6.3	RandomWalk2dMobilityModel	22
2.6.4	RandomRectanglePositionAllocator	23
3	Popis konfiguračního souboru	24
3.1	Konstrukce konfiguračního souboru	28
3.2	Aplikace konfiguračního souboru do modelu	29
3.3	Výhody konfiguračního souboru	29
4	Vývoj webového rozhraní	30
4.1	Prostředky pro vývoj uživatelského rozhraní	30
4.1.1	Co je to PHP	30
4.1.2	Instalace prostředí pro vývoj v PHP.....	30
4.1.3	Psaní skriptů PHP	31
4.1.4	Spouštění skriptů PHP	31
4.1.5	Nette Framework	31
4.1.6	Jazyk HTML	32
4.1.7	Kaskádové styly CSS.....	32
4.1.8	JavaScript.....	32
4.2	Návrh uživatelského rozhraní	32
4.2.1	Hlavička webu	34
4.2.2	Tělo webu	34
4.2.3	Informační panel	35
4.2.4	Patička webu	35
5	Implementace rozhraní pro generování konfiguračního souboru	36
5.1	HTML stránka.....	36
5.2	Provázání s Nette Framework	37
5.3	Prvky uživatelského rozhraní.....	38
5.3.1	Vytvoření formuláře	38
5.3.2	Textová pole	38
5.3.3	Roletková menu	39
5.3.4	Odesílací tlačítko	39
5.3.5	Zobrazování volitelných formulářů	40

5.3.6	Vykreslení formuláře	40
5.3.7	Validace formuláře podle závislosti vyplněných hodnot	41
5.4	Generování souboru a jeho stažení	41
5.4.1	Uložení hodnot z formulářů	41
5.4.2	Stažení souboru	42
5.5	Testování generátoru	43
5.5.1	Kontrola W3C validátory	45
5.6	Používání vytvořeného webového rozhraní	45
6	Závěr	47
	Literatura	48
	Seznam použitých zkratk	50
	Seznam příloh	51
A	Schéma modelu	52
B	Diagram funkce webové aplikace generátoru	53
C	Webová stránka generátoru	54
C.1	Konfigurační část	54
C.2	Informační text o úspěšném vygenerování	55
D	Obsah přiloženého DVD	56
E	Návod ke spuštění webu generátoru na lokálním webovém serveru	57

SEZNAM TABULEK

Tab. 3.1: Parametry konfigurace pro běžný chod simulace	24
Tab. 3.2: Parametry konfigurace pro nastavení přenosu dat přes FTP	25
Tab. 3.3: Parametry konfigurace pro nastavení VOIP komunikace	25
Tab. 3.4: Parametry konfigurace pro nastavení počátečního umístění uzlů 2D modelů	26
Tab. 3.5: Parametry konfigurace pro nastavení počátečního umístění uzlů 3D modelů	26
Tab. 3.6: Parametry konfigurace pro nastavení parametrů modelů mobility	27
Tab. 3.7: Parametry konfigurace pro nastavení param. GaussMarkov. modelu mobility	27

ÚVOD

Bezdrátové sítě jako prostředek pro komunikaci používáme běžně. Denně používáme Wi-Fi sítě pro připojení našich notebooků, chytrých telefonů i tabletů k různým sítím, které mají ale jedno společné. Touto společnou věcí je užití centrálního přístupového bodu. A právě přístupový bod, může být pro některé bezdrátové sítě největší komplikací. Dosah sítě pro všechny účastníky je totiž limitovaný vzdáleností od tohoto, případně od více těchto přístupových bodů. Tento fakt znemožňuje užití těchto pro nás běžně známých Wi-Fi sítí ve velkých oblastech, v oblastech bez infrastruktury a podobně.

Již v roce 1973 Defense Advanced Research Projects Agency (DARPA) zahájila výzkumy bezdrátového přenosu dat, motivované potřebou poskytnout počítačovou bezdrátovou komunikaci v mobilním prostředí. V takovém prostředí kde není kabelová ani jiná bezdrátová infrastruktura a kde není výstavba takovéto infrastruktury výhodná. Byla to prostředí např. bitevních polí, oblasti zasažené katastrofami apod. A právě jejich výsledkem, Manet sítě se bude tato bakalářská práce zabývat. Praktická část bude probíhat se síťovým simulátorem Ns-3, který je cílený především pro výzkumné a vzdělávací účely. Výsledkem mé práce bude webová aplikace pro generování konfiguračního souborů, která bude usnadňovat nastavení modelu sítě Manet.

V první kapitole naleznete vznik a popis Manet sítí, rozdělení a přehled používaných směrovacích protokolů. Obsahuje také stručný popis několika z nich. Druhá kapitola zahrnuje popis síťového simulátoru Ns-3, důležité abstrakce pro práci s ním a různé modely mobility, které budou použity v konfiguračním souboru. Třetí kapitola popisuje vznik mého konfiguračního souboru, jeho obsah a práci s ním. Čtvrtá kapitola se zabývá vznikem mého webového rozhraní. Obsahuje popis všech použitých technologií a ovládacích prvků generátoru konfiguračního souboru. V páté kapitole je uveden popis programovacího kódu všech částí webového rozhraní a informace o testování generátoru. Naleznete v ní také popis používání výsledné webové aplikace z pohledu uživatele.

Bakalářská práce obsahuje přehledně zpracované přílohy, které obsahují schéma mého modelu, diagram funkce webové aplikace generátoru, snímky obrazovek webového generátoru, obsah přiloženého DVD a návod jak spustit web generátoru na lokálním webovém serveru.

1 MANET SÍTĚ

1.1 Co jsou a jak vznikly

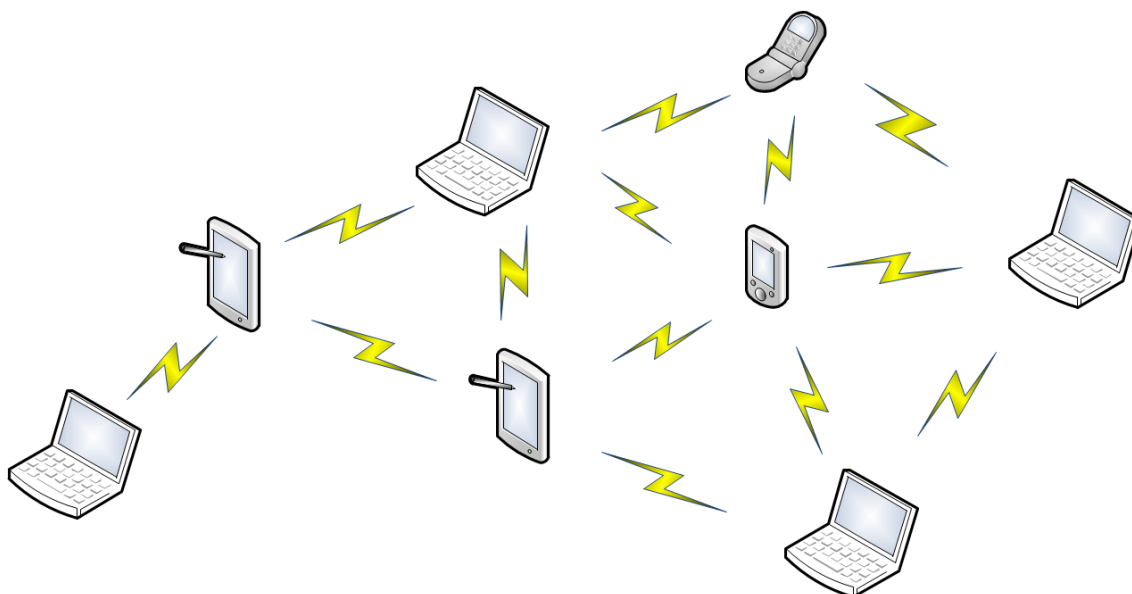
Manet sítě mají již svou mnoha letou historii. V roce 1973 Defense Advanced Research Projects Agency (DARPA) zahájila výzkum přepínání paketů u tzv. store and forward rádiové komunikace pro spolehlivou komunikaci mezi počítači. DARPA Packet Radio Network (PRNET) vyvinuly v průběhu let robustní, spolehlivou a funkční experimentální síť. PRNET poskytovala přes rádiový kanál výměnu dat mezi počítači, které byly geograficky oddělené. Takovéto komunikační médium poskytovalo pro uživatele několik důležitých výhod. Jednou z výhod je mobilita, jelikož síť PRNET může pracovat za pohybu stanic. Druhou výhodou byla snadná instalace bez potřeby kabeláže. Mezi další výhody patří snadné přidávání stanic a přenastavení sítě [1].

Toto byl začátek pro sítě, které dnes označujeme Manet, což je zkratka Mobile ad-hoc network, o kterých pojednává tato bakalářská práce.

Store and forward rádiová komunikace: Český přepojování zpráv je označení jednoho ze způsobu dopravy zpráv v sítích, především v případě, že odesílatel nemůže přímo komunikovat s příjemcem. Při této dopravě zpráv je paket dočasně uložen ve vyrovnávací paměti a zkontrolován pomocí pole CRC. Pokud je zcela bez chyb je přeposlán dál, jinak je vyřazen. Toto přepojování zpráv umožňuje odesílání paketů mezi různě rychlými sítěmi [2].

1.2 Mobile ad hoc síť

Jsou takové mobilní sítě, které nevyužívají žádnou pevnou infrastrukturu. Uzly (nodes) v ad hoc síti jsou propojeny navzájem způsobem peer-to-peer, tedy přímo bez centrálního prvku. Schéma této sítě je zobrazeno na obrázku 1.1. Centrální prvky jako jsou směrovače, přepínače, přístupové body v sítích tohoto typu nejsou používány. Každý uzel v této síti musí zastávat funkci směrovače. Tyto sítě bývají používány pro speciální, často improvizované služby přizpůsobené daným aplikacím. Ad hoc síť může být sestavena pro omezený čas. Komunikační protokoly jsou laděny pro specifické účely aplikací (např. video přenos skrze bitevní pole, záchranné práce, senzorové čidla, dopravní komunikace a jiné). Topologie sítě a aplikace na daných sítích se může dynamicky měnit. Díky tomu je pro správnou funkci této sítě důležitou vlastností samo-konfigurace. Vzhledem k mobilnímu užití i v jinak špatně přístupných oblastech je důležitá nízká spotřeba elektřiny uzlů. Velkou roli při použití této sítě hraje taktéž bezpečnost, díky její možné rozlehlosti a bezpečnostním účelům. Dalšími jejich charakteristickými vlastnostmi jsou omezená šířka pásma a proměnná kapacita spojení [3][4].



Obr. 1.1: Manet síť

1.3 Dělení protokolů pro Manet síť

Díky výše uvedeným vlastnostem Manet sítí a jejich různorodému použití jsou potřeba různé druhy směrovacích protokolů, které jsou tříděny do několika kategorií.

Směrovací protokoly Manet sítí dělíme takto:

V závislosti na způsobu aktualizace směrovací trasy:

- Záplavově založené algoritmy (Flooding based algorithms): Tyto algoritmy užívají záplavu směrovacích aktualizací v síti. Díky tomu se zvyšuje režie v síti. Na této metodě je založena většina protokolů (AODV, DSR, GSR a jiné).
- Algoritmy zpětné linky (Link reversal algorithms): Tyto algoritmy šetří šířku pásma. Záznam o výpadku slouží zároveň k vytvoření náhradních cest. Algoritmus dává k dispozici mnoho takovýchto alternativních cest. Tímto nejen šetří šířku pásma, ale také poskytuje odchozí cesty v případě výpadků linky.

V závislosti na více uzlové komunikaci:

- Unicasting – jedno směrové: Zahrnují všechny protokoly, které jsou použity pro komunikaci z bodu do bodu.
- Multicasting – více směrové: Zahrnují všechny protokoly, které jsou použity pro komunikaci z jednoho bodu do více bodů v síti. To jsou takové, kdy jeden uzel rozesílá data celé skupině.

- Geocasting – geografické: Zahrnují všechny protokoly, které jsou použity pro geografické účely.

Další způsob jak třídit směrovací protokoly může být založený na síťových hierarchiích - strukturách:

- Rovinatý: Všechny uzly jsou v síti na stejné úrovni.
- Hierarchický: Uzly jsou logicky nebo fyzicky na rozdílné úrovni dle struktury.

Více směrové protokoly jsou tříděny v těchto dvou skupinách v závislosti na topologii:

- Stromově založené (tree-based): V této topologii existuje pouze jedna možná cesta mezi zdrojovým a cílovým uzlem. Stromové protokoly jsou dále děleny do sdílené stromové a zdrojové stromové topologie.
- Síťově založené (mesh-based): Protokoly jsou více odolné proti změnám v síti. V této topologii může existovat více možných cest mezi zdrojovým a cílovým uzlem.

Jsou různé kategorie, jak můžeme řadit MANET protokoly. Hlavní kategorie mohou být rozděleny na proaktivní (řízené tabulkou), reaktivní (dle požadavku) a hybridní (kombinující oba dva tyto mechanismy) [4]. Přehled směrovacích protokolů pro Manet sítě naleznete na obrázku 1.2.

1.3.1 Proaktivní směrovací protokoly

Při proaktivních směrovacích protokolech uzly pravidelně kontrolují síť pro možnosti směrování. Uzly si mezi sebou vyměňují informace o topologii a tímto si vždy udržují aktuální směrovací tabulku. Výhodou je, že dané uzly znají cestu v okamžiku její potřeby. Z tohoto vyplývá nízké zpoždění. Nevýhodou je omezení přenosové šířky pásma směrovacími informacemi, případně možné zahlcení sítě těmito informacemi [4]. Přehled těchto protokolů naleznete na obrázku 1.2.

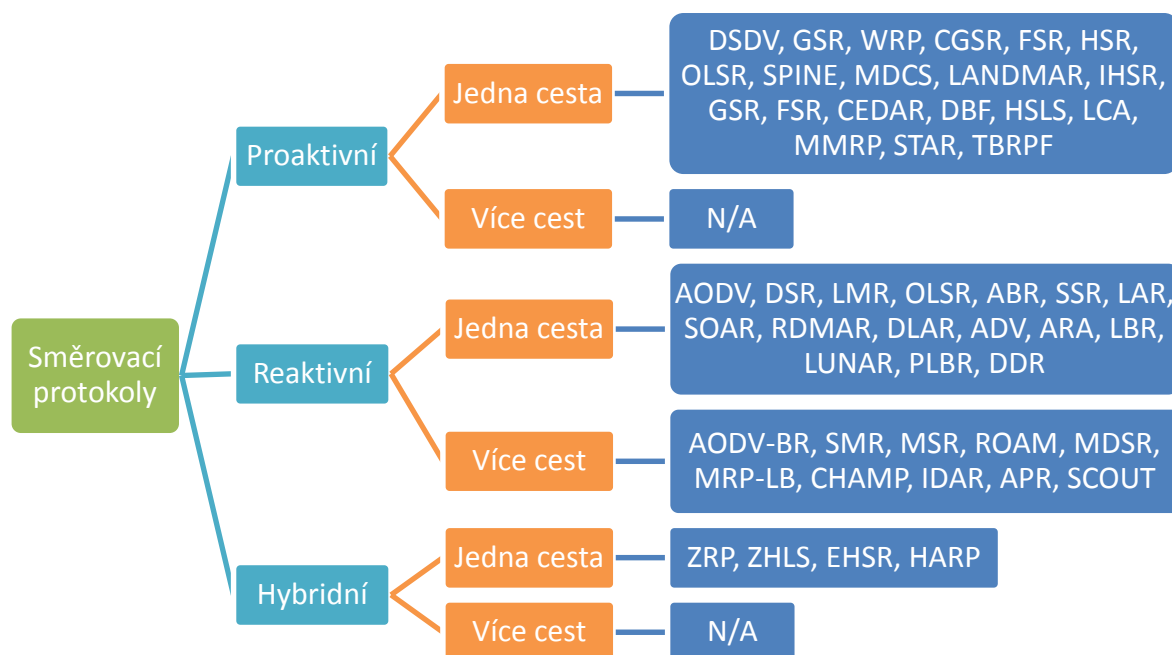
1.3.2 Reaktivní směrovací protokoly

Reaktivní protokoly představují skutečnou povahu ad hoc sítě, které jsou mnohem dynamičtější než klasické infrastrukturní sítě. Místo pravidelné aktualizace, která je u proaktivních protokolů, tyto protokoly aktualizují směrovací informace, až když vznikne požadavek na směrování. Výsledkem je tedy menší zatížení sítě směrovacími informacemi. Tato skutečnost má největší význam v sítích s velkou mobilitou uzlů [4]. U těchto protokolů vyplývá nevýhoda většího zpoždění, než u proaktivních protokolů, způsobená hledáním cesty až v okamžiku její potřeby. Přehled těchto protokolů naleznete na obrázku 1.2.

1.3.3 Hybridní směrovací protokoly

Hybridní protokoly jsou kombinací dvou výše popsaných typů. Proaktivní strategie je používána pro objevení a udržení spoje u blízkých uzlů, zatímco spoje vzdálenější jsou

objevy a udržovány reaktivní strategií. Díky tomu jsou negativní vlastnosti, jako je velké množství režijních informací u proaktivních protokolů a zpoždění u reaktivních protokolů navzájem minimalizovány [4].



Obr. 1.2: Ukázka dělení směrovacích protokolů [4]

1.4 Nejvýznamnější směrovací protokoly pro MANET sítě

Z velkého množství protokolů vybírám pro stručný popis následující protokoly. Dva protokoly proaktivní – OLSR, DSDV, dva protokoly reaktivní – DSR, AODV a jeden protokol hybridní – ZRP.

1.4.1 Protokol OLSR

Jedná se o optimalizovanou verzi tradičního link state protokolu OSPF vyvíjenou pro MANET sítě. Jedná se o protokol proaktivní. Charakteristickým rysem tohoto protokolu je jeho koncept takzvaných mnohabodových distribučních přenosů (MPRs = multipoint distribution relays). OLSR používá dva typy zpráv a to zprávu „hello“ a „TC“ (Topology Change). Jedná se o dobře škálovatelný algoritmus a hodí se pro velké i malé bezdrátové sítě [3][4].

Směrovací protokoly Link-State: Přidělují váhové koeficienty linkám na základě jejich parametrů. Optimalizují trasy pomocí součtu daných metrik a výběrem dané trasy s nejmenším součtem [6].

1.4.2 Protokol DSDV

Jedním z dalších proaktivních protokolů je protokol DSDV. Algoritmus DSDV je založen na algoritmu protokolu RIP, který byl modifikován pro možnou pohyblivost uzlů. Směrovací

tabulky mají položky s množstvím přeskoků ke každému cílovému uzlu, každá položka je označena pořadovým číslem. Uzly pravidelně komunikují se směrovací tabulkou jejich sousedů a také komunikují, když je dostupná nová informace. Směrovací informace jsou přenášeny použitím všesměrových vysílání a vícebodových vysílání [4].

1.4.3 Protokol DSR

Jedná se o protokol reaktivní. Je jednoduchý a robustní, navrhnutý pro užití ve více skokové bezdrátové ad-hoc síti. Byl navrhnut pro užití v síti kde je až 200 uzlů. Každý uzel v DSR je zodpovědný za potvrzení, který další uzel ve zdrojové cestě přijímá pakety. Dodatečná režie směrování se vyskytuje pouze tehdy, když vznikne změna v topologii (v statických sítích se směrovací informace vyskytují pouze z počátku). U velkých sítí bude ve směrovací tabulce velké množství cest a mnohé z nich budou zastaralé. Neaktuální směrovací vyrovnávací paměť může mít za následek nesrovnalosti při tvorbě nové cesty. Další nevýhodou je to, že sestavovací zpoždění spojení je vyšší než u proaktivních protokolů. Proto protokol funguje dobře ve statických a málo měnitelných prostředích, v rychle se měnících prostředích se výkon degraduje [4].

1.4.4 Protokol AODV

Tento protokol je další ukázkou protokolů reaktivních. Podporuje více skokové směrování mezi mobilními uzly pro založení i údržbu sítě. AODV je založený na protokolu DSDV. Protokol používá pořadová čísla k sledování čerstvosti daných informací. Jednou z výhod AODV je to, že se pokouší minimalizovat množství požadovaných všesměrových vysílání [4].

1.4.5 Protokol ZRP

Jedná se o protokol ze skupiny hybridních. Kombinuje tedy vlastnosti proaktivních a reaktivních protokolů. Tento protokol řadí uzly do podsítí. Uvnitř daných podsítí pracuje s proaktivním směrováním, mimo tyto podsítě používá reaktivní směrování. Tento protokol je díky hybridnímu směrování vhodný pro širokou škálu užití, zvláště u velkých sítí s různorodým chováním pohyblivosti [4].

2 SIMULÁTOR NS-3

2.1 Co je to simulátor Ns-3

Jedná se o diskretními událostmi řízený síťový simulátor, cílený především pro výzkumné a vzdělávací účely. Ns-3 je zdarma, licencovaný GNU GPLv2 licencí a veřejně dostupný. Hlavní cíle projektu Ns-3 jsou vytvořit uživateli preferované, otevřené prostředí pro výzkum sítí. Ns-3 je následovník projektu Network simulator 2, není ale jeho rozšířením, jedná se o nový simulátor.

Ns-3 projekt se zavázal k budování silného simulačního jádra, které je dobře zdokumentované, snadno použitelné a laditelné. Ns-3 podporuje rozvoj simulačních modelů, které jsou dostatečně realistické, tak aby je bylo možno použít v reálném čase v propojení se skutečnými sítěmi. Ns-3 simulační jádro podporuje výzkum na sítích založených na IP protokolu, ale i na sítích jiného typu. Velká většina uživatelů se zaměřuje na bezdrátové IP simulace, které zahrnují modely pro Wi-Fi, WiMAX, LTE nebo na simulace na fyzické a linkové vrstvě a na různé statické nebo dynamické směrovací protokoly.

Ns-3 umožňuje propojení s reálnými systémy. Uživatelé mohou například přijímat a vysílat pakety generované v simulátoru na skutečných síťových zařízeních. Ns-3 může také sloužit jako propojovací konstrukce mezi virtuálními stroji, aby vytvořila různé vlastnosti přenosového média [7].

2.2 Výhody oproti jiným simulátorům

Ve srovnání s ostatními obdobnými simulátory se Ns-3 vyznačuje následujícími výhodami a rozdíly na úrovni návrhu.

- Mnoho simulátorů používá specifické modelovací jazyky. Ns-3 používá C++ nebo Python a umožňuje plnou podporu daného jazyka.
- Ns-3 obsahuje nízko úroňové rozhraní pro programování (API), které umožňuje pokročilým uživatelům flexibilní konfiguraci různými způsoby. Nad tímto rozhraním můžou uživatelé používat tzv. pomocníky (helpers), kteří umožňují snadnější používání funkcí. Uživatelé tedy mají možnost volby, zda budou vytvářet simulace na nízké úrovni API, která jim umožní větší flexibilitu, nebo dají přednost jednodušší práci s pomocníky. Programovou organizaci Ns-3 si můžete prohlédnout na obrázku [2.1](#).
- Návrh simulace je zaměřen na možnost propojení s reálnými sítěmi. Uzly v Ns-3 jsou vytvořeny dle vzoru Linuxové architektury. Klíčová rozhraní a objekty jsou obdobné s těmi, které jsou v Linuxovém počítači.

- Události simulace v Ns-3 jsou jednoduše vyvolané funkční procedury, které jsou naplánované, aby vykonávaly simulace v předepsaném čase. Do události může být vytvořena každá funkce.
- Rozdílem je taktéž chybějící standardní vývojové prostředí. Chybějící standardní vývojové prostředí umožňuje uživatelům pracovat v příkazové řádce a integrovat nástroje např. pro vizualizace dle své potřeby. Uživatel má tedy svobodu v tom, jakým způsobem, nebo v jakém prostředí simulace bude tvořit. Někteří vývojáři používají vývojové prostředí Eclipse.

test				
pomocníci (helpers)				
protokoly	aplikace	zařízení	rozšiřování	...
internetové modely		mobilní modely		
sít'				
jádro				

Obr. 2.1: Programová organizace Ns-3 [8]

2.3 Spouštění simulací v Ns-3

Existuje více způsobů jak spustit simulaci v Ns-3. Pomocí příkazů v konzoli je nejjednodušší následující:

```
./waf --run „nazev-simulace“
```

Je potřeba být ve složce s danou simulací, případně její cestu uvést do názvu simulace. Dále je nutné, aby ve složce, ve které zadáváme příkaz, byl spustitelný soubor WAF. Soubor WAF obsahuje možnosti, které automaticky aktualizují cestu k Python a najdou si Ns-3 modul [8].

V prostředí které je používáno při tvorbě bakalářské práce, stačí pro spuštění simulace stisknout klávesovou zkratku CTR + B (Build) a výsledek se v Eclipse zobrazí v konzoli. V tomto případě je potřeba ale pro danou simulaci mít dobře nastavenou cestu k Build složce a příkazu. Vlastní projekty se nacházely ve složce Scratch. Ve složce např. mezi příklady z tutoriálů, nefungovalo korektně sestavování simulací z aktuálních souborů. Pro práci na bakalářské práci je používána verze simulátoru Ns-3.14.

2.4 Používané abstrakce

Před prací v Ns-3 je potřeba se seznámit s několika abstrakcemi v systému.

2.4.1 Uzel

V abstrakci Ns-3 se pojmem uzel označuje jakékoliv přístrojové zařízení. V internetových sítích se běžně používá označení host, nebo koncové zařízení. Jelikož ale není Ns-3 pouze internetový simulátor, bylo zvoleno označení uzel. Tato abstrakce je v C++ označovaná třídou Node.

2.4.2 Aplikace

Je označení pro uživatelský program, které běží na daných uzlech a generuje nějakou aktivitu. V C++ je označena třídou Application. Příklady aplikací jsou UdpClientServer, UdpEcho a jiné.

2.4.3 Kanál

Představuje propojení uzlu k objektu. V C++ je zastoupen třídou Channel. Tato třída poskytuje metody pro vytvoření komunikačního kanálu.

2.4.4 Síťové zařízení

V simulátoru nahrazuje zařízení, která v počítačové terminologii označujeme síťovou kartou. Abstrakce tohoto zařízení zahrnuje jak softwarový ovladač, tak simulované zařízení. V C++ je označeno jako třída NetDevice.

2.4.5 Pomocníci topologie (helpers)

Pro usnadnění konfigurace velkého množství uzlů a kanálů se v Ns-3 používají tzv. pomocníci. Ti nám umožňují nastavovat různé vlastnosti prvků v co nejjednodušší formě.

2.5 Definování mobility v Ns-3

Podpora mobility v Ns-3 zahrnuje:

- Sadu modelů, které se používají ke sledování a udržování aktuální pozice v kartézské soustavě a aktuální rychlosti objektu.
- Objekty pro zachycení událostí.
- Pomocné třídy, které se používají pro umístění uzlů a modely nastavení mobility.

Ns-3 používá k určení polohy objektů kartézský souřadný systém [7].

2.6 Modely k nastavení mobility uzlů

V Ns-3 existují následující modely definující mobilitu uzlů v simulaci [5]:

- BuildingsMobilityModel,
- ConstantAccelerationMobilityModel,

- ConstantPositionMobilityModel,
- ConstantVelocityMobilityModel,
- GaussMarkovMobilityModel,
- HierarchicalMobilityModel,
- RandomDirection2dMobilityModel,
- RandomWalk2dMobilityModel,
- RandomWaypointMobilityModel,
- SteadyStateRandomWaypointMobilityModel,
- WaypointMobilityModel.

V následujících podkapitolách budou některé z těchto modelů popsány, jelikož jsou použity v mém konfiguračním souboru.

2.6.1 GaussMarkovMobilityModel

Jedná se o model 3D pohybu. Na rozdíl od většiny jiných modelů má tento model paměť, v které si pamatuje předchozí parametry a podle nich generuje nové. Pole pohybu je omezeno 3D ohraničujícím rámečkem o tvaru kvádra. Tento model se hodí například pro jednoduchý let letadla [5].

Parametry modelu:

- 3D meze oblasti (Bounds) v kterých bude probíhat pohyb.
- Časovač změn (TimeStep) – změna aktuálního směru a rychlosti po dané době pohybu.
- Alfa (Alpha) – konstanta reprezentující laditelný parametr v tomto modelu.
- Střední rychlost (MeanVelocity) – náhodná proměnná k přiřazení průměrné rychlosti.
- Střední směr (MeanDirection) – náhodná proměnná k přiřazení průměrného směru.
- Střední výška (MeanPitch) – náhodná proměnná k přiřazení průměrné výšky.
- Normální rychlost (NormalVelocity) – Gaussova náhodná proměnná pro výpočet další hodnoty rychlosti.
- Normální směr (NormalDirection) – Gaussova náhodná proměnná pro výpočet další hodnoty směru.
- Normální výška (NormalPitch) – Gaussova náhodná proměnná pro výpočet další hodnoty výšky.

Příklad použití:

```
mobility.SetMobilityModel ("ns3::GaussMarkovMobilityModel",
    "Bounds", BoxValue (Box (-100, 100, 0, 100, 0, 0)),
    "TimeStep", TimeValue (Seconds (0.1)),
    "Alpha", DoubleValue (1),
    "MeanVelocity", RandomVariableValue (UniformVariable (0,
    1)),
    "MeanDirection", RandomVariableValue (UniformVariable (0,
```

```

6.28319)),
"MeanPitch", RandomVariableValue (UniformVariable (0, 0)),
"NormalVelocity", RandomVariableValue (NormalVariable
(0.0, 1, 10)),
"NormalDirection", RandomVariableValue (NormalVariable
(0.0, 1, 10)),
"NormalPitch", RandomVariableValue (NormalVariable (0.0,
1, 10)));

```

2.6.2 RandomDirection2dMobilityModel

Pohyb uzlů je založen na náhodných směrech. Každý objekt se pozastaví na specifickou dobu, vybere si náhodný směr a rychlost a pohybuje se v tomto směru, dokud nenarazí na jednu z hranic. Když hranice dosáhne, opakuje výběr náhodného směru a rychlosti [5].

Parametry modelu jsou:

- 2D meze oblasti (Bounds) v kterých bude probíhat pohyb.
- Rychlost (Speed) – náhodná proměnná k ovládání rychlosti pohybu uzlů v m/s.
- Pozastavení (Pause) – náhodná proměnná k ovládání přestávky. Proměnou hodnotou je čas v sekundách, který definuje, v jakých intervalech bude pozastavení fungovat.

Příklad použití:

```

mobility.SetMobilityModel
("ns3::RandomDirection2dMobilityModel",
"Bounds", RectangleValue (Rectangle (0, 20, 0, 20)),
"Speed", RandomVariableValue (ConstantVariable (2)),
"Pause", RandomVariableValue (ConstantVariable (0.2)));

```

2.6.3 RandomWalk2dMobilityModel

Každý prvek se pohybuje rychlostí a směrem zvoleným náhodně podle uživatelových veličin. Pohybuje se, dokud nenarazí na meze oblasti nebo po určitou dobu [5].

Parametry modelu jsou následující:

- Meze oblasti (Bounds) v kterých bude probíhat pohyb.
- Čas (Time) mezi změnou aktuálního směru a rychlosti.
- Vzdálenost (Distance) o kterou se má prvek posunout před změnou aktuálního směru a rychlosti.
- Režim (Mode) který označuje podmínku používanou ke změně aktuálního směru a rychlosti.
- Rychlost (Speed) – náhodná proměnná k ovládání rychlosti pohybu uzlů v m/s.

- Směr (Direction) – proměnná k definování směru pohybu. Směr se udává v gradientech.

Příklad použití:

```
mobility.SetMobilityModel "ns3::RandomWalk2dMobilityModel",
    "Bounds", RectangleValue (Rectangle (-100, 100, -100,
    100)),
    "Speed", RandomVariableValue (ConstantVariable (1)),
    "Time", TimeValue (NanoSeconds (1000));
```

Z kódu lze dle popisu výše vyčíst, v jaké oblasti bude probíhat pohyb, že rychlost pohybu prvků bude 2 m/s a čas mezi změnami rychlosti a směru bude 1 000 ns.

2.6.4 RandomRectanglePositionAllocator

Každý uzel si náhodně vybere bod a do něho se náhodnou rychlostí a s náhodnými pauzami pohybuje. Až dorazí do daného bodu tak proces výběru bodu a pohybu se opakuje [5].

Parametry modelu jsou:

- Rychlost (Speed) – náhodná proměnná k ovládání rychlosti pohybu uzlů v m/s.
- Pozastavení (Pause) – náhodná proměnná k ovládání přestávky. Proměnou hodnotou je čas v sekundách, který definuje, v jakých intervalech bude pozastavení fungovat.
- Alokátor pozice (PositionAllocator) – náhodně generuje z nastaveného rozsahu umístění uzlů.

Příklad použití:

```
ObjectFactory pos;
pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
pos.Set ("X", RandomVariableValue (UniformVariable (10, 30)));
pos.Set ("Y", RandomVariableValue (UniformVariable (5, 15)));

Ptr<PositionAllocator> taPositionAlloc = pos.Create ()-
>GetObject<PositionAllocator> ();
mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
    "Speed", RandomVariableValue (ConstantVariable (2)),
    "Pause", RandomVariableValue (ConstantVariable (2)),
    "PositionAllocator", PointerValue (taPositionAlloc));
mobility.SetPositionAllocator (taPositionAlloc);
```

3 POPIS KONFIGURAČNÍHO SOUBORU

V této kapitole je popsán můj konfigurační soubor. V tabulkách jsou zpracovány všechny parametry konfigurace, které můj konfigurační soubor obsahuje s vysvětlením, k čemu slouží. Tabulky jsou rozděleny podle oblasti použití daných nastavení.

Tab. 3.1: Parametry konfigurace pro běžný chod simulace

Parametr	Význam
typProtokolu	Nastavení použitého směrovacího protokolu v modelu. Lze použít tyto možnosti: 1 – AODV 2 – DSVD 3 – OLSR
typKomunikace	Nastavení použitého typu komunikace na síti. Lze použít tyto možnosti: 1 – VOIP 2 – FTP 3 – VOIP i FTP
zvolenaMobilita	Nastavení použitého typu mobility. Lze použít tyto možnosti: 1 – RandomWalk2dMobilityModel 2 – RandomDirection2dMobilityModel 3 – RandomRectanglePositionAllocator 4 – GaussMarkovMobilityModel
pocetUzlu	Nastavení počtu uzlů.
adresy	Začátek adresního prostoru pro přiřazování IP adres.
maska	Nastavení masky sítě.
casUkonceniSimulace	Doba, po které se ukončí simulace (s).
SSID	SSID bezdrátové sítě.
nazevAnimSouboru	Volba názvu pro vytvoření animačního souboru, typicky s příponou .xml.
statistikyToku	Název souboru pro statistiky toku dat.
intervalTiskuSmerTabulky	Pravidelný interval tisku směrovací tabulky do souboru (s).

Tab. 3.2: Parametry konfigurace pro nastavení přenosu dat přes FTP

Parametr	Význam
ftpZuzlu	Uzel, z kterého se budou přenášet data přes FTP.
ftpNaUzel	Uzel, na který se budou přenášet data přes FTP.
ftpAplikaceStart	Čas začátku aplikace pro přenos FTP (s).
ftpAplikaceKonec	Čas konce aplikace pro přenos FTP (s).
maxPrenesenychBytuFTP	Maximální množství dat pro přenos skrz FTP (B).
ftpVelikostPaketu	Velikost jednoho paketu pro přenos skrz FTP (B).
casZacatkuSink	Čas začátku aplikace pro příjem dat FTP (s).
casKonceSink	Čas konce aplikace pro příjem dat FTP (s).
portFTP	Nastavení portu FTP.

Tab. 3.3: Parametry konfigurace pro nastavení VOIP komunikace

Parametr	Význam
voipZuzlu, voip2Zuzlu	Nastavení uzlu, z kterého bude probíhat VOIP komunikace.
voipNaUzel	Nastavení uzlu, na který bude probíhat VOIP komunikace.
casZacatkuVoipAplikace	Čas začátku VOIP aplikace na daném uzlu (s).
casKonceVoipAplikace	Čas ukončení VOIP aplikace na daném uzlu (s).
casZacatkuVoipKlient	Čas začátku příjmu VOIP aplikace na klientovi (s).
casKonceVoipKlient	Čas konce příjmu VOIP aplikace na klientovi (s).
voipVelikostPaketu	Velikost jednoho paketu pro VOIP (B).
rychlostPrenosuDatVOIP	Rychlost přenosu VOIP (Kb/s).
AccessClassVOIP	Nastavení priority služby pro případ použití QOS.
casZapnutiVOIP	Doba, po kterou je zapnutá komunikace VOIP (s).

casVypnutiVOIP	Doba, po kterou je vypnutá komunikace VOIP (s).
portVOIP	Nastavení portu VOIP.

Tab. 3.4: Parametry konfigurace pro nastavení počátečního umístění uzlů 2D modelů

Parametr	Význam
MinX	Souřadnice X prvního objektu.
MinY	Souřadnice Y prvního objektu.
DeltaX	Mezera mezi dalšími objekty, souřadnice X.
DeltaY	Mezera mezi dalšími objekty, souřadnice Y.
GridWidth	Počet objektů na jednom řádku.

Tab. 3.5: Parametry konfigurace pro nastavení počátečního umístění uzlů 3D modelů

Parametr	Význam
osaXmin	Minimální krajní hodnota osy X pro generování náhodné pozice objektu.
osaXmax	Maximální krajní hodnota osy X pro generování náhodné pozice objektu.
osaYmin	Minimální krajní hodnota osy Y pro generování náhodné pozice objektu.
osaYmax	Maximální krajní hodnota osy Y pro generování náhodné pozice objektu.
osaZmin	Minimální krajní hodnota osy Z pro generování náhodné pozice objektu.
osaZmax	Maximální krajní hodnota osy Z pro generování náhodné pozice objektu.

Tab. 3.6: Parametry konfigurace pro nastavení parametrů modelů mobilit

Parametr	Význam
oblastXmin, oblastXmax	Krajní hodnoty osy X oblasti, ve které bude probíhat pohyb.
oblastYmin, oblastYmax	Krajní hodnoty osy Y oblasti, ve které bude probíhat pohyb.
rychlost	Rychlost pohybu uzlů.
casPredZmenou	Nastavení času mezi změnami rychlostí a směru (ns).
smerPohybu	Směr pohybu v radiánech.
pauza	Nastavení v jakých intervalech bude v pohybu uzlů pauza (s).
nahodnaXod, nahodnaXdo	Hranice osy X, mezi kterými bude probíhat pohyb u RandomWaypointMobilityModel.
nahodnaYod, nahodnaYdo	Hranice osy Y, mezi kterými bude probíhat pohyb u RandomWaypointMobilityModel.

Tab. 3.7: Parametry konfigurace pro nastavení parametrů GaussMarkovMobilityModel modelu mobility

Parametr	Význam
boxXmin, boxXmax	Hranice osy X 3D oblasti ve tvaru kvádru, ohraňující oblast pohybu u modelu GaussMarkovMobilityModel.
boxYmin, boxYmax	Hranice osy Y 3D oblasti ve tvaru kvádru, ohraňující oblast pohybu u modelu GaussMarkovMobilityModel.
boxZmin, boxZmax	Hranice osy Z 3D oblasti ve tvaru kvádru, ohraňující oblast pohybu u modelu GaussMarkovMobilityModel.
casovacZmen	Doba, po které se má změnit aktuální směr a rychlost pohybu (s).
Alfa	Konstanta reprezentující laditelný parametr u modelu GaussMarkovMobilityModel.
strRychlostMin, strRychlostMax	Hranice hodnot pro vygenerování průměrné rychlosti.

strSmerOd, strSmerDo	Hranice hodnot pro vygenerování průměrného směru.
strVyskaOd, strVyskaDo	Hranice hodnot pro vygenerování průměrného výšky.
normRychlostPrumer normSmerPrumer normVyskaPrumer	Střední hodnoty pro normální (Gaussovo) rozdělení.
normRychlostRozptyl normSmerRozptyl normVyskaRozptyl	Hodnota rozptylu pro normální (Gaussovo) rozdělení.
normRychlostHranice normSmerHranice normVyskaHranice	Hraniční hodnoty pro normální (Gaussovo) rozdělení.

3.1 Konstrukce konfiguračního souboru

Při psaní konfiguračního souboru byly použity takové datové typy, které byly popsány v dokumentacích k daným funkcím. Parametry konfiguračního souboru byly zpracovány do hlavičkového souboru s příponou .h. Pro možnosti výběru protokolu, druhu mobility i druhu komunikace byl použit ve zdrojovém kódu modelu příkaz pro větvení programu `switch`. V případě zadání volby, která není definována, se v konzoli vypíše upozornění. Schéma mého modelu je v příloze A. Níže je uvedena ukázka příkazu `switch` pro výběr protokolu.

```
switch (typProtokolu)
{
    case 1:
    {
        DsdvHelper dsdv;
        Ipv4StaticRoutingHelper staticRouting;
        InternetStackHelper internet;
        internet.SetRoutingHelper (dsdv);
        internet.Install (wifinodes);
        Ptr<OutputStreamWrapper> routingStream =
            Create<OutputStreamWrapper> ("DSDV.routes",
            std::ios::out);
        dsdv.PrintRoutingTableAllEvery (Seconds
            (intervalTiskuSmerTabulky), routingStream);
    } break;
    case 2:
    {
        OlsrHelper olsr;
        Ipv4StaticRoutingHelper staticRouting;
        Ipv4ListRoutingHelper list;
        list.Add (staticRouting, 0);
        list.Add (olsr, 100);
    }
}
```

```

    InternetStackHelper internet;
    internet.SetRoutingHelper (list);
    internet.Install (wifinodes);
    Ptr<OutputStreamWrapper> routingStream =
    Create<OutputStreamWrapper> ("OLSR.routes",
    std::ios::out);
    olsr.PrintRoutingTableAllEvery (Seconds
    (intervalTiskuSmerTabulky), routingStream);
} break;
case 3:
{
    AodvHelper aodv;

    ...

} break;
default: printf("\nNeplatná volba pro výběr
protokolu.\n\n"); break;
}

```

3.2 Aplikace konfiguračního souboru do modelu

Pro použití konfiguračního souboru je potřeba ho mít přiřazen ve zdrojovém kódu modelu následujícím přiřazujícím příkazem:

```
#include "konfigurace.h"
```

Konfigurační soubor s názvem `konfigurace.h` musí být ve složce s daným zdrojovým kódem modelu nebo je potřeba připsat do přiřazujícího příkazu relativní cestu k adresáři, kde se nachází. Části kódu, které tvoří daný směrovací protokol, síťový provoz a druh pohybu uzlů musí být obsaženy ve zdrojovém kódu modelu.

3.3 Výhody konfiguračního souboru

Díky vytvořenému konfiguračnímu souboru, lze jednoduše nastavovat parametry simulovaného modelu a to z jediného centrálního souboru. Výhodou je, že uživatel pro změny parametrů nemusí znát zdrojový kód modelu. Pouze změni hodnoty přehledně pojmenovaných proměnných, v souboru `konfigurace.h`, které jsou v této práci popsány. Po spuštění simulace v simulátoru Ns-3, bude daná simulace provedena s těmito parametry.

4 VÝVOJ WEBOVÉHO ROZHRAŇÍ

Pokračováním této bakalářské práce je vývoj webového rozhraní pro generování mého konfiguračního souboru. Webovým rozhraním se zabývá 4. a 5. kapitola. V následujících podkapitolách naleznete stručný popis prostředků a technologií použitých pro vytvoření uživatelského rozhraní a popis rozložení rozhraní.

4.1 Prostředky pro vývoj uživatelského rozhraní

Pro vytvoření tohoto rozhraní byl zvolen programovací jazyk PHP s využitím frameworku Nette verze 2.0.10. Základ webové stránky, do které jsou vloženy PHP skripty, je vytvořen v jazyce HTML s použitím kaskádových stylů CSS. Některé vlastnosti níže popsaného frameworku Nette využívají skriptovacího jazyka JavaScript.

4.1.1 Co je to PHP

PHP je široce používaný mnohoúčelový skriptovací jazyk, který je obzvláště vhodný pro vývoj webových aplikací a lze jej vložit do HTML. Hlavním cílem tohoto jazyka je umožnit webovým vývojářům rychle psát dynamicky generované stránky. Vznik jazyku PHP se datuje od roku 1994, kdy byl vytvořen Rasmussem Lerdorfem. Při psaní této práce (v březnu 2013) je nejnovější stabilní verze jazyka PHP 5.4.12. V současné době používá jazyk PHP více než 200 milionů webových stránek po celém světě [10].

4.1.2 Instalace prostředí pro vývoj v PHP

Pro spouštění skriptů na počítači je potřeba mít nainstalován webový server a prostředí PHP. Uživatel má možnost si tyto potřebné programy nainstalovat a poté nastavit sám, případně lze použít jeden z mnoha předpřipravených instalačních balíčků. Předpřipravené instalační balíčky obsahují všechny potřebné programy a často některé další, které usnadní programování v PHP, případně jsou jinak užitečné. Takovýto balíček ušetří uživateli čas a některé starosti s nastavováním programů. Pro operační systém Windows se jedná například o tyto balíčky: Complex Web Server, XAMPP, EasyPHP. Všechny tyto balíčky byly mnou nainstalovány a vyzkoušeny. Každý se trochu liší ve výsledném používání.

Pro práci na bakalářské práci bylo zvoleno prostředí instalované z balíčku EasyPHP. Balíček EasyPHP obsahuje tyto programy: PHP, Apache HTTP Server Project, MySQL, PhpMyAdmin, Xdebug. Aplikace Apache HTTP Server Project slouží k vytvoření webového serveru na vlastním počítači, díky němu mohou být PHP skripty spouštěny na lokálním počítači. MySQL je databázový serverový systém. Aplikace PhpMyAdmin slouží ke správě MySQL databázového systému pomocí webového prohlížeče. Xdebug je rozšíření umožňující ladění PHP skriptů [11].

Pro zapnutí webového serveru na vlastním počítači je potřeba spustit aplikaci EasyPHP, které se následně projeví zobrazením ikony v oznamovací oblasti. Pomocí této ikony lze spustit konfiguraci všech nástrojů.

4.1.3 Psaní skriptů PHP

Pro práci skriptů v PHP lze použít obyčejný textový editor. Pro přehlednější práci je vhodné použít editor, který zvýrazňuje a například i doplňuje syntaxi. Při tvorbě bakalářské práce byl používán textový editor PSPad. PSPad umožňuje například také ještě kompresi kódu, automatické opravy a jiné.

4.1.4 Spouštění skriptů PHP

Standardní umístění souborů s PHP skripty na počítači s instalačním balíčkem EasyPHP je adresář „C:\Program Files (x86)\EasyPHP-12.1\www“. Internetové stránky z toho adresáře lze spustit z internetového prohlížeče zadáním IP adresy „http://127.0.0.1“ což znamená adresu našeho počítače, tzv. localhost. Je možné rovnou do adresního řádku zadat localhost.

PHP je serverová technologie, veškeré PHP skripty se provedou na serveru, na kterém jsou uloženy dané stránky. Uživatel se již zobrazí pouze výsledná podoba stránek v HTML bez skriptů. Výhoda serverové technologie je v tom, že dané skripty nejsou závislé na použitém internetovém prohlížeči ani jiném prostředí v klientském počítači. Taktéž nezáleží na technickém vybavení počítače klienta. Další výhodou je, že klient nemůže zobrazit zdrojový kód skriptů.

4.1.5 Nette Framework

Pojmem framework se označují programové struktury, které slouží jako opora při programování a vývoji softwarových projektů. Framework obsahuje především nějakou sadu již naprogramovaných kódů a dále může obsahovat různé podpůrné programy. Používání takového dobré programové struktury usnadní a zrychlí práci, zpřehlední kód a taktéž eliminuje bezpečnostní rizika při psaní vlastních složitých kódů. Pomocí frameworku jsme schopni vytvořit robustnější aplikace v kratším čase než při použití čistého kódu. Eliminace bezpečnostních rizik je díky tomu, že tyto rozhraní bývají programována často velkou komunitou programátorů, kteří dokážou dané části kódu ošetřit lépe než jeden programátor. Díky použití frameworku nemusí programátor zbytečně věnovat svůj čas opakujícím se úkolům a různým drobnostem, které odvádějí od programování pozornost.

Pro programování v PHP existuje velké množství frameworků. Toto je způsobenou velkou rozšířeností jazyka PHP. Frameworky se liší v rozsahu obsažených skriptů, množství a kvality dokumentací a taktéž ve výsledné rychlosti s jakou dokáží na serverech pracovat. Jedná se například o framework Nette, Zend, Cakephp, Qcodo a spousty dalších [12].

Nette Framework je framework pro vytváření webových aplikací v PHP 5. Jedná se o český framework jehož zakladatelem je David Grudl. Vznikl v roce 2004 a od roku 2008 byl zpřístupněn veřejnosti. Framework je možné zdarma používat i v komerčních projektech. Nette Framework obsahuje velké množství kvalitní dokumentace v českém jazyce. Jedná se o Framework s nejrozsáhlejší aktivní komunitou u nás. Tento framework používá velké množství i velkých webových stránek a portálů v České republice [13].

4.1.6 Jazyk HTML

První specifikace jazyka HTML byla uvolněna v roce 1991 z Evropské organizace pro jaderný výzkum (CERN), kde na ní od roku 1989 pracoval vynálezce Tim Berners-Lee. Tento jazyk původně vznikl pro komunikaci mezi vědci a organizacemi ze vzdálených míst po světě [14]. Jazyk HTML je tzv. značkovací jazyk. Mezi HTML tagy (značky) se uzavírají části dokumentů a tímto se jim dává přesný význam, který zobrazí internetový prohlížeč. Většina HTML tagů jsou párové, otevírají se pomocí značek zapsaných v ostrých závorkách < > a uzavírají se znakem lomítka za první ostrou závorkou </ >. V současné době probíhá vývoj jazyka HTML verze 5. Některé jeho vlastnosti již moderní internetové prohlížeče podporují. Některé z těchto vlastností jsou ve webové stránce generátoru použity.

4.1.7 Kaskádové styly CSS

Kaskádové styly jsou nadstavbou jazyka HTML, určené pro nastavení vzhledu prvků webové stránky. Neovlivňují obsah dokumentu. Jejím hlavním smyslem bylo oddělit nastavení vzhledu dokumentu od jeho struktury a obsahu. Díky užití CSS se dají nastavit větší možnosti podoby webu, než s pouhým jazykem HTML. Jeho velkou výhodou je možnost definice stylů v jednom samostatném souboru, který může nastavovat vzhled celého webu.

4.1.8 JavaScript

JavaScript je skriptovací programovací jazyk, jehož kód je proveden ve webovém prohlížeči na straně klienta. JavaScript dovoluje reagovat na různé události na webu. Ve webu generátoru je použit tento skriptovací jazyk pro validaci formulářových prvků.

4.2 Návrh uživatelského rozhraní

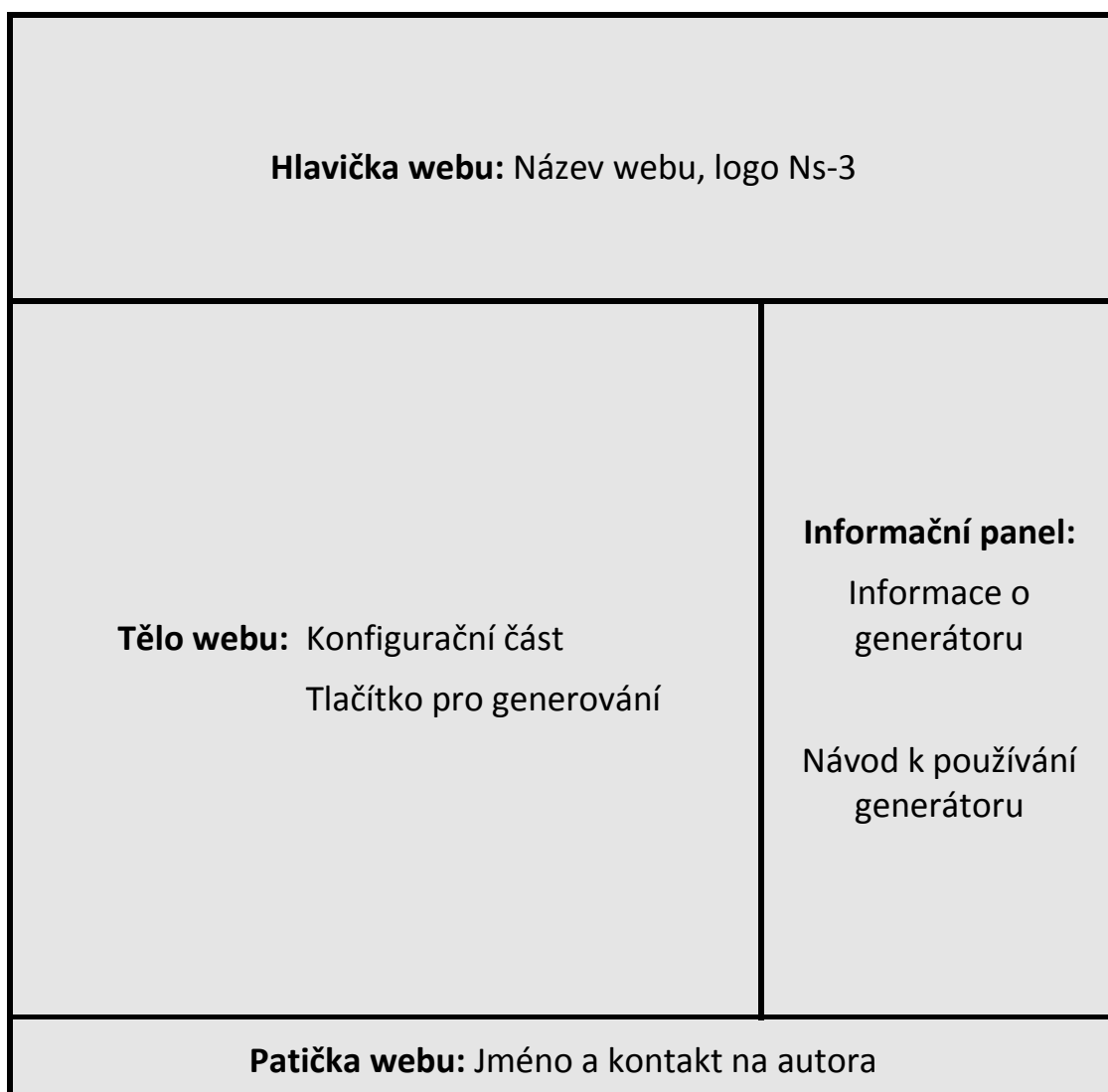
Internetová stránka mého generátoru konfiguračního souboru je vytvořena v jednoduchém a přehledném formátu. Hlavní cíl stránky je vytvořit konfigurační soubor pro návštěvníky zabývající se simulacemi Manet sítí v simulátoru Ns-3, nejdůležitější je tedy plná funkčnost stránek a snadné používání.

Tato webová stránka musí být zcela kompatibilní se všemi běžně používanými internetovými prohlížeči. Šířka stránky musí být zvolena tak, aby stránka byla dobře použitelná

i na monitorech s menším rozlišením. Vzhledem k tomuto důvodu byla zvolena šířka stránky 850 px. Na následujícím obrázku 4.1 je zobrazeno rozvržení stránky mého generátoru.

Webové rozhraní generátoru je vytvořeno ze dvou webových stránek, na kterých není potřebné žádné menu. Po odeslání formuláře se pouze v těle webu místo formuláře zobrazí informační text s možností návratu zpět na nastavení. Takovéto provedení je velmi přehledné a snadno použitelné, uživatel se na webu dobře zorientuje. Barvy webové stránky částečně odpovídají barvám loga simulátoru Ns-3. Pozadí webu, použité písmo a další barvy jsou vybrány pro dobrou čitelnost a přehlednost obsahu.

Základ webové stránky je vytvořen v jazyce HTML. Rozčlenění stránky na jednotlivé části je vytvořeno pomocí HTML elementů `<div>` s kombinací kaskádových stylů CSS.



Obr. 4.1: Rozvržení internetové stránky generátoru

4.2.1 Hlavička webu

V hlavičce webové stránky je umístěn název stránky a logo simulátoru Ns-3, které je volně dostupné na webových stránkách simulátoru. Barva písma názvu webové stránky je shodná se zelenou barvou obsaženou v logu Ns-3.

4.2.2 Tělo webu

V těle webu je konfigurační část, kde si uživatel navolí požadované vlastnosti. Pod konfigurační částí je umístěno tlačítko pro vytvoření konfiguračního souboru.

Pro zadávání hodnot konfiguračního souboru na webu slouží následující komponenty:

- textová pole,
- roletková menu (tzv. selectbox).

Pomocí textových polí uživatel zadává například počet uzlů v síti, délku trvání simulace ale i rozsah IP adres simulovaných uzlů. U nejnovějších internetových prohlížečů s podporou HTML 5 jsou u textových polí s číselnými hodnotami vyobrazeny šipky pro nastavování hodnot. Vzhled takového textového pole je ukázán na obrázku 4.2. Při kliknutí na popis textového pole se kurzor automaticky objeví ve formulářovém políčku.

Textová pole jsou ošetřena proti zadávání dat v nesprávném formátu nebo rozsahu. Není tedy možné odeslat např. IP adresu v neplatném formátu. Uživatel je na špatný formát dat upozorněn varováním, které se zobrazí pod textovým polem, ihned při špatně zadané hodnotě. V případě že uživatel zadá hodnotu, která je ve správném tvaru, ale v návaznosti na další hodnoty polí by ji nešlo v simulátoru použít, je uživatel upozorněn varováním na začátku webové stránky. Toto varování se zobrazí po stisku tlačítka pro vygenerování konfiguračního souboru.



Obr. 4.2: Nový typ číselného textového pole dle HTML 5

Roletková menu slouží na webu k výběru směrovacího protokolu, druhu komunikace a typu použitého pohybového modelu. Výhodou roletkového menu je to, že uživatel si může vybrat pouze z předem daných možností, které má náš model naprogramovány.

Pro vygenerování konfiguračního souboru je pod výše popsány konfiguračními prvky umístěno tlačítko s popisem „Generovat konfigurační soubor“. V případě že jsou hodnoty ve formuláři vyplněny správně, se po stisku tohoto tlačítka zobrazí text o úspěšném vygenerování souboru a soubor se automaticky nabídne ke stažení. Text je zobrazen místo formuláře. Z této

stránky lze vygenerovaný konfigurační soubor stáhnout taktéž ručně, nebo lze přejít zpět na nastavení formuláře. Vzhled stránky po úspěšném vygenerování souboru je v příloze C.2.

Dané prvky v těle webu jsou přehledně pojmenované. Při načtení stránky jsou prvky předvyplněny základními hodnotami, které uživateli usnadní vytváření konfiguračního souboru.

4.2.3 Informační panel

Tento panel je zobrazen na pravé straně těla webu. Obsahuje instrukce ke generátoru a informace o daném webu. Jedná se o krátké popisy, proto jsou zakomponovány do jediné stránky vedle konfigurační části. Graficky jsou odděleny, aby nerušily, ale uživatel si je může pohodlně přečíst a dozví se například, že web byl vytvořen v rámci bakalářské práce. Jako jediný oddíl v generátoru je pomocí CSS umístěn absolutně s přesně nastavenou šířkou i délkou.

4.2.4 Patička webu

V patičce webové stránky se nachází jméno autora a kontaktní emailová adresa. Tyto údaje jsou zobrazeny pod sebou, zarovnány doprava. Emailová adresa je zapsána formátem pro užití protokolu `mailto`, který po kliknutí na emailovou adresu spustí poštovního klienta s předpřipraveným předmětem emailu.

5 IMPLEMENTACE ROZHRANÍ PRO GENEROVÁNÍ KONFIGURAČNÍHO SOUBORU

V následujících podkapitolách je uvedeno, jak je naprogramována stránka generátoru a její ovládací prvky. V poslední podkapitole jsou popsány problémy objevené při testování.

5.1 HTML stránka

Internetová stránka generátoru je rozvržena pomocí blokového HTML elementu `<div>`, který je určen pro vyčlenění oddílu na stránce. Ve stránce je použit v kombinaci s kaskádovými styly, které definují vlastnosti daného oddílu. Kaskádové styly jsou připojeny ke stránce generátoru z externího souboru `style.css` následujícím příkazem:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Každý element `<div>` má svůj identifikátor `id`, podle kterého se propojí s vytvořenými kaskádovým styly. V kaskádových stylech jsou poté pro každou část stránky nastaveny parametry pro správné vykreslení bloku. Všechny oddíly na stránce jsou obaleny jedním hlavním oddílem, který určuje šířku celé stránky a zarovnání na střed obrazovky.

Oddíly v CSS nejsou umístěny absolutní polohou, ale jsou umístěny pomocí obtékání, díky tomu nemusí mít přesně danou velikost a dokáží se přizpůsobit proměnné velikosti formuláře. Absolutně je umístěn pouze informační panel a to tak, že překrývá prázdnou část oddílu, která obaluje tělo webu.

Parametry oddílu těla jsou tedy obtékání směrem vlevo, pevná šířka, barva pozadí a nastavení oddělovací přerušované čáry. Dále je nastavena minimální výška těla, která má vliv na stránku po odeslání formuláře. Taktéž jsou definovány fonty písma, kterých je ve stylech zapsáno několik, pro případ, že uživatel v počítači nejvíce preferované nemá.

```
#obaltela {
    float: left;
    width: 850px;
    background-color: #9BA5A5;
}

#telo {
    float: left;
    width: 616px;
    min-height: 700px;
    border-right: 4px;
    border-right-style: dashed;
    border-right-color: #89969B;
    background-color: #9BA5A5;
```

```
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
}
```

Informativní text o generátoru je vypsán pomocí klasického HTML elementu pro odstavce, kterým je:

```
<p> </p>
```

PHP kódy v generátoru jsou vloženy přímo do kódu HTML stránky. PHP kód je ohraničen těmito příkazy:

```
<?php ?>
```

5.2 Provázání s Nette Framework

Jelikož pro vytvoření dané webové stránky používám framework Nette, je potřeba jej se stránkou propojit. V adresáři s webovou stránkou v počítači i na následném serveru je nutné mít uložené soubory s daným frameworkem. Framework Nette je vytvořen ve dvou formách. První forma je užívaná při programování aplikací a druhá zminimalizovaná forma je určená pro nahrání na server. Rozsáhlá forma tohoto frameworku verze 2.0.10 obsahuje 240 souborů ve 40 složkách. Výhodou této verze je logické umístění souborů a jejich vložené komentáře. Minimalizovaná forma je tvořena pouze jedním souborem `nett.min.php`. Tato verze byla vytvořena sloučením všech souborů a odstraněním komentářů a mezer. Funkčně jsou tyto dvě formy stejné, avšak minimalizovaná forma byla vytvořena pro finální použití již na serveru s danou webovou aplikací. Výhoda, která z ní vyplývá, je vyšší rychlost frameworku na serveru a jednoduché nahrání pouze jediného souboru na server [13].

Ve zdrojovém kódu webové aplikace je nutné provést propojení se soubory obsahující framework, které provedeme jednoduchým následujícím příkazem.

```
require 'Nette/loader.php';
```

Soubor `loader.php` se již postará a zavedení potřebných tříd. V případě použití minimalizované formy frameworku, se v souboru `loader.php` nachází již kompletní framework.

Dále je ve zdrojovém kódu potřeba napsat příkazy, které stanovují jaké části Nette frameworku jsou v kódu zahrnuty.

```
use Nette\Forms\Form;
use Nette\Utils\Html;
use Nette\Utils\SafeStream;
use Nette\Utils\Strings;
```

Tyto příkazy znamenají, že je použita programová struktura pro formuláře, pro práci s HTML elementy, protokol SafeStream pro atomické operace se soubory a třída pro práci s řetězci.

Nette pro některé akce používá JavaScript. Používá ho například pro validaci formulářů na uživatelské straně. Díky JavaScriptu vypíše okno, nebo hlášku upozorňující na chybu ve formuláři. Taktéž díky němu funguje na stránce generátoru zobrazování různých skupin nastavení, dle typu komunikace. Web generátoru používá vylepšený JavaScriptový kód, který obsahuje tzv. živou validaci. Oproti standartní validaci upozorňuje na chyby v textových polích ihned. A to rovnou při psaní, varovnou zprávou pod polem. Propojení stránky s JavaScriptovým kódem Nette je provedeno tímto příkazem:

```
<script src="live-form-validation.js"></script>
```

5.3 Prvky uživatelského rozhraní

5.3.1 Vytvoření formuláře

Při vytváření formuláře je uveden pouze jeho název. V generátoru není použita žádná odesílací metoda dat.

```
$form = new Form;
```

5.3.2 Textová pole

Následujícím zdrojovým kódem je vytvořeno textové pole, které má textový popis „Počet uzů“ a jeho hodnota je dostupná pod identifikátorem `pocetUzlu`. Díky příkazu `setRequired` bude uživatel upozorněn na prázdné textové pole vypsáním upozornění.

Pomocí příkazu `addRule` se u formulářů dají nastavit různé validační pravidla. U textových polí v generátoru je používána podmínka `Form::INTEGER`, která vyjadřuje, že obsah pole může být pouze celé číslo. Dále je používaná podmínka `Form::RANGE`, která slouží k omezení obsahu polí na daný číselný rozsah hodnot. Je použita v polích nastavujících čas, aby nemohly být záporné a u počtu uzlů, pro nastavení minimálně dvou uzlů. U textového pole pro zadávání IP adresy a masky je použita podmínka `Form::PATTERN`, která stanovuje validaci podle zadaného regulárního výrazu, díky kterému se kontroluje, zda IP adresa nebo maska má reálný tvar. Dany regulární výraz pochází ze zdroje [16]. Podmínkou `Form::MAX_LENGTH` je nastaveno u SSID velikost maximálně 31 znaků, kterou akceptuje Ns-3 simulátor.

Příkaz `setType` určuje, jaký typ dat formulář obsahuje, čehož je využito u HTML verze 5, pro zobrazení vylepšeného textového pole. V prohlížeči podporující HTML 5 se zobrazí malé šipky pro nastavování čísel.

```

$form->addText('pocetUzlu', 'Počet uzlů:')
    ->setRequired('Zadejte prosím počet uzlů')
    ->addRule(Form::INTEGER, 'Počet uzlů musí být celé číslo.')
    ->addRule(Form::RANGE, 'Jsou nutné minimálně dva uzly.', array(2, NULL))
    ->setType('number');

```

Při načtení stránky jsou textová pole předvyplněná základními hodnotami, které uživateli usnadní vytváření konfiguračního souboru. Tyto základní hodnoty jsou doplněny do textových polí pomocí příkazu `setDefaults`, obsahující pole s danými hodnotami a názvy textových polí.

```

$form->setDefaults(array(
    'pocetUzlu' => '10',
    'casUkonceniSimulace' => '30',
    ...
));

```

5.3.3 Roletková menu

Nabízejí uživateli předpřipravené možnosti volby a v Nette se vytvářejí pomocí příkazu `addSelect`. Jeho parametry jsou název `typProtokolu`, textový popis „Směrovací protokol:“ a položky které toto menu obsahuje. Je možné vypsát položky přímo do příkazu `addSelect` nebo uvést v příkazu název pole, které dané položky obsahuje. Ve stránce generátoru je použito zadání formou pole, které je přehlednější a snadno změnitelné. Při načtení stránky je vybrána první položka z pole.

```

$typProtokolu = array(
    '1' => 'AODV',
    '2' => 'DSVD',
    '3' => 'OLSR',
);
$form->addSelect('typProtokolu', 'Směrovací protokol:',
    $typProtokolu);

```

5.3.4 Odesílací tlačítko

Tlačítko, po jehož stisku se vygeneruje konfigurační soubor, je tvořeno následujícím kódem. Tlačítko se jmenuje `send` a má daný popis.

```

$form->addSubmit('send', 'Generovat konfigurační soubor');

```

5.3.5 Zobrazování volitelných formulářů

Dle výběru komunikace v modelu z roletkového menu se zobrazují upřesňující nastavení pro VOIP, FTP nebo VOIP i FTP komunikaci. Každá daná komunikace má ve formuláři vytvořenou svojí skupinu, která je při načtení stránky skryta a zobrazuje se pomocí metody `toggle`. Při výběru dané volby z roletkového menu, tato metoda předává nastavený argument `ID`, díky kterému se poté skryté formuláře zobrazí. Zobrazování a skrývání formulářů se provádí JavaScriptem. Jelikož na stránce jsou použity skupiny formulářů VOIP i FTP u dvou možností v roletkovém menu, jsou tomuto podmínky pro metodu `toggle` přizpůsobeny. Jsou zadány formou pole, s výčtem voleb kdy mají být dané `ID` předány.

```
$form->addSelect('typKomunikace', 'Typ komunikace:',  
$typKomunikacePole)  
->addCondition(Form::IS_IN, array(1, 3))  
->toggle("V")  
->endCondition()  
->addCondition(Form::IS_IN, array(2, 3))  
->toggle("F")  
->endCondition();
```

Skupiny formulářů jsou poté vytvořeny následujícím způsobem. Do HTML elementu `<fieldset>`, který vytváří skupiny formulářů, se vloží parametr `ID` a nastavení pro skrytí skupiny formulářů. V případě, že by v HTML elementu nebylo nastaveno základní skrytí, skupiny formulářů by při načítání stránky problikly.

```
$form->addGroup("Nastavení VOIP")->setOption('container',  
Html::el('fieldset')->id('V')->style("display:none"));
```

5.3.6 Vykreslení formuláře

Automatické vykreslení formuláře v Nette obstarává tzv. `DefaultFormRenderer`. Ten je ale ve stránce generátoru trochu upraven pro vylepšený vzhled. HTML elementu `<fieldset>` je přiřazen parametr `align="center"` pro zarovnání názvu skupin formulářů na střed. Zarovnání není provedeno pomocí CSS, jelikož se jedná u kaskádových stylů o problematické nastavení, díky velmi špatné kompatibilitě u různých webových prohlížečů. Obdobným způsobem je nastavena třída CSS stylu popisu prvků formuláře. Příkazem `wrappers` se upravují dané HTML elementy, nebo se také dají vytvořit zcela nové.

```
$renderer = $form->getRenderer();  
$renderer->wrappers['group']['label'] = 'legend  
align=center';  
$renderer->wrappers['label']['container'] = 'th  
class=label_poli';
```


Formulář je výsledně zobrazen v HTML stránce pomocí příkazu echo, který se v jazyce PHP používá pro výstup na obrazovku.

```
echo $form;
```

5.3.7 Validace formuláře podle závislosti vyplněných hodnot

Jelikož je nemožné spouštět například FTP aplikaci z uzlu, který neexistuje, nebo ji zapínat v čase po konci simulace, má můj generátor ošetřeny i tyto závislosti. Tato validace se provádí ve funkci `onValidate[]`, která se spustí až po vyplnění formuláře a po odeslání hodnot tlačítkem pro vygenerování konfiguračního souboru. V případě, že jsou ve formuláři chyby, vypíší se konkrétní varování nad formulář a vygenerování konfiguračního souboru se neprovede. Po té co uživatel chyby opraví, lze pokračovat v generování konfiguračního souboru. Aby byla funkce `onValidate[]` funkční, je potřeba volat ve formuláři funkci `fireEvents()`, která slouží pro obsluhu událostí, reagujících na odesílací tlačítko.

```
$form->fireEvents();

$form->onValidate[] = function ($form) {
    $form->cleanErrors();
    if($form['pocetUzlu']->value < $form['ftpZuzlu']->value)
    { $form->addError('Číslo uzlu z kterého probíhá FTP
      přenos musí být menší než počet uzlů
      (<'. $form['pocetUzlu']->value.').'); }
    };
};
```

5.4 Generování souboru a jeho stažení

V následujících podkapitolách jsou uvedeny postupy, které jsou prováděny při generování konfiguračního souboru, jeho zápisu do textového souboru a nabídnutí ke stažení.

5.4.1 Uložení hodnot z formulářů

V případě odeslání validního formuláře jsou data z formuláře zapsána do textového souboru. Zápis hodnot se provádí v těle podmínky pro úspěšné odeslání.

```
if ($form->isSuccess()) { ... }
```

Hodnoty z formulářových prvků jsou předány do pole `$values` metodou `getValue()`. Metoda `getValue()` navíc z textových polí automaticky odstraní zbytečné mezery vlevo i vpravo.

```
$values = $form->getValues(TRUE);
```

Nyní lze již získaná data z pole zapisovat do textového souboru. Je ovšem důležité, aby operace pro zápis do souboru probíhaly atomicky a izolovaně. V jiném případě by mohly vzniknout chyby při užívání generátoru více uživateli zároveň. K tomuto používá Nette Framework protokol SafeStream, díky němuž lze atomicky manipulovat se soubory prostřednictvím běžných PHP funkcí. Protokol SafeStream zapisuje data prvně do pomocného souboru a až při ukončení zápisu se vytvoří skutečný soubor.

Pro vytváření a otevírání souborů se používá funkce `fopen`, jejichž parametry jsou cesta k souboru (před kterou je parametr `safe`, značící užití protokolu SafeStream) a parametr určující druh přístupu k souboru, který je zde `w`. Parametr `w` stanovuje, že se soubor otevře pro zápis, v případě že neexistuje, bude vytvořen, v případě že již existuje, bude přepsán novým souborem. Ukazatel na daný soubor je zaznamenán do proměnné `$soubor`, s kterou se dále pracuje.

```
$soubor = fopen('safe://konfigurace.h', 'w');
```

Zápis dat do souboru se provádí pomocí funkce `fwrite`, s parametry ukazatel na soubor a textový řetězec, určený pro zápis do souboru. Textové řetězce se v jazyce PHP spojují pomocí tečky. Do souboru je zapsán datový typ proměnné, název proměnné a hodnota získána z pole `$values`. V případě zápisu hodnoty SSID je funkcí `Strings::toAscii()` odstraněna případná diakritika. Následuje tečkou spojený další řetězec, který ve formě konstanty obsahuje středník a znaky pro nové řádky. Pro konce těchto řádků byly vytvořeny konstanty příkazem `define`, z důvodů větší přehlednosti a úspory místa. Do textového souboru jsou taktéž zapsány všechny ostatní části konfiguračního souboru, které uživatel v generátoru nenastavuje [15].

```
define('NL', ";\r\n\r\n");  
fwrite $soubor, "int  
    typProtokolu=".$values['typProtokolu'].NL);
```

Po dokončení zápisu dat se soubor uzavře funkcí `fclose`. Díky použití protokolu SafeStream se až nyní všechny data zapíší do výsledného souboru na serveru.

```
fclose($soubor);
```

5.4.2 Stažení souboru

Po úspěšném odeslání formuláře se zobrazí nová stránka informující o úspěšném vygenerování souboru, který se nabídne ke stažení. Tato stránka je zobrazeno v příloze C.2.

Přesměrování na tuto stránku je provedeno funkcí `header` s parametrem `Location`, která předá informace o přesměrování stránky přímo v hlavičce HTTP protokolu.

```
$form->onSuccess[] = function ()
{
    header('Location: stazeni_souboru.html');
};
```

Webová stránka `stazeni_souboru.html` obsahuje informace o úspěšném provedení vygenerování konfiguračního souboru, možnost stáhnout vygenerovaný soubor a možnost návratu na předchozí stránku s nastavením. Je vytvořena pouze v jazyce HTML, neobsahuje žádný PHP kód. Automatické stahování konfiguračního souboru po zobrazení stránky je způsobeno jejím přesměrováním pomocí meta tagu na PHP skript, který nemá žádný výstup k zobrazení, pouze vyvolá přenos souboru.

```
<meta http-equiv="refresh" content="0;url=ulozsoubor.php">
```

Skript v souboru `ulozsoubor.php` obsahuje níže uvedené příkazy, které zajišťují nucené stáhnutí nastaveného souboru. Využívá funkce `header` a to s několika parametry. `Content-Description` je popis obsahu a je nastaven na přenos souborů. `Content-Type` stanovuje stažení souboru. V parametru `Content-Disposition` se určí soubor ke stažení. Příkazy `ob_clean()` a `flush()` slouží k vymazání a vyprázdnění výstupního zásobníku. Příkaz `readfile` přečte soubor a zapíše do výstupního zásobníku. Příkazem `exit` je nakonec ukončena sekvence příkazů [15].

```
header("Content-Description: File Transfer");
header("Content-Type: application/force-download");
header("Content-Disposition: attachment;
        filename=\"konfigurace.h\"");
ob_clean();
flush();
readfile ("konfigurace.h");
exit;
```

5.5 Testování generátoru

Během programování generátoru a po jeho dokončení byly jeho vlastnosti testovány v následujících prohlížečích:

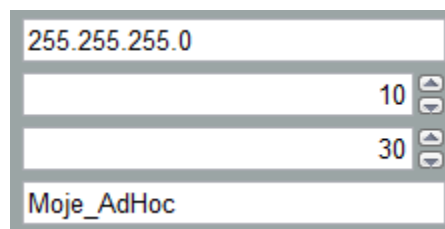
- Google Chrome (verze 26.0.1410.64),
- Internet Explorer (verze 10.0.9200.16540),
- Mozilla Firefox (verze 21.0),
- Safari (verze 5.1.7.),

- Opera (verze 12.15).

Jelikož technologie PHP je serverová a úkony pro generování konfiguračního souboru jsou zpracovávány na serveru, tak se jeho funkčnost v různých prohlížečích neliší. Jediné problémy které se při testování objevovaly, byly se zachováním stejného vzhledu stránky ve všech prohlížečích. Tyto problémy jsou způsobovány rozdílnou interpretací některých parametrů CSS stylů v různých prohlížečích. Největší rozdíly se objevovaly v prohlížeči Opera a jsou dále popsány a ukázány.

Nejvýraznější rozdíl byl u prohlížeče Opera v zarovnávání obsahu číselných polí formuláře vpravo, zatímco ostatní pole byla zarovnávána vlevo. Přehledně je to vidět na obrázku [5.1](#), přímo ze stránky generátoru. Toto chování změnil následující CSS styl nastavující všem vstupním polím zarovnání vlevo.

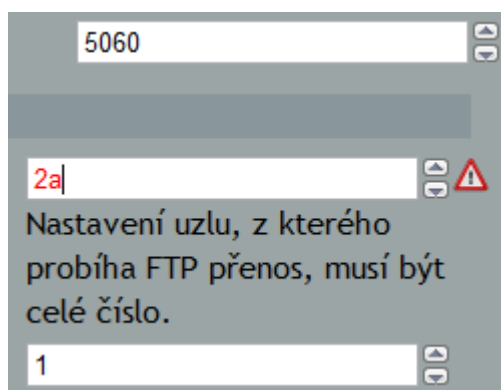
```
input
{
text-align: left;
...
}
```



Obr. 5.1: Rozdílné zarovnání obsahu textových polí

Další nepříjemnou vlastností Opery bylo rozdílné zobrazování výstražné ikony u chybně vyplněného pole. Opera ji zobrazila ihned za pravou stranu pole, posunula celou skupinu polí do leva, oproti ostatním a zúžila popis textového pole. Problém je zobrazen na obrázku [5.2](#). Rozdílné chování bylo odstraněno CSS stylem, nastavujícím zobrazování chybové zprávy do bloku.

```
.form-error-message
{
...
display: block;
}
```



Obr. 5.2: Chybné zobrazování výstražné ikony

5.5.1 Kontrola W3C validátory

Webová stránka generátoru i CSS styly byly kontrolovány internetovými validátory mezinárodního konsorcia W3C (World Wide Web Consortium).

U hlavní webové stránky jsou nalezeny 4 chyby. První z nich je způsobena chováním Nette frameworku, který u formuláře automaticky vytvoří parametr `action`, ale bez hodnoty, jelikož na webu generátoru není použit. Další 3 chyby jsou způsobeny zastaralostí parametrů `align`. Tento parametr byl použit úmyslně u názvu skupin formulářů, kvůli kompatibilitě mezi prohlížeči a situace je popsána v kapitole 5.3.6. Stránka zobrazující informační text o úspěšném vygenerování neobsahuje žádnou chybu.

CSS styly jsou plně validní, bez jediné chyby nebo varování.

5.6 Používání vytvořeného webového rozhraní

Stránka webového rozhraní generátoru konfiguračního souboru je při načtení vyplněna základními hodnotami. Vzhled stránky konfigurační části je zobrazen v příloze C.1.

Uživatel si ve vytvořeném webovém rozhraní generátoru konfiguračního souboru může pomocí tří roletkových menu vybrat typ směrovacího protokolu, druh komunikace a použitou mobilitu uzlů. Dle typu vybrané komunikace, se zobrazují dvě skupiny formulářových polí s upřesňujícími nastaveními pro komunikaci VOIP, FTP nebo jejich kombinaci. Všechny ostatní parametry uživatel mění v textových polích, které mají přesně definován formát vstupních hodnot a rozsah, pokud to typ parametru vyžaduje. V případě špatně zadané hodnoty je uživatel ihned informován varováním, pod špatně vyplněným textovým polem, které po opravě chyby ihned zmizí. U textových polí s číselnou hodnotou jsou v nejnovějších internetových prohlížečích zobrazeny šipky pro snadnější nastavování hodnot. V textových

polích, kam uživatel zadává textové znaky, může používat diakritiku, jelikož před zápisem těchto hodnot do konfiguračního souboru se diakritika odstraní.

Po té co uživatel vyplní prvky formuláře dle jeho požadavků, si může již nechat vygenerovat konfigurační soubor stiskem tlačítka v dolní části stránky s názvem „Generovat konfigurační soubor“. Webová stránka ověří, zda zvolená nastavení můžou být v simulaci použita. Pokud se stane, že uživatel nastaví například start VOIP aplikace z uzlu, který nebude kvůli zvolenému menšímu počtu uzlů v simulaci vůbec vytvořen, generování konfiguračního souboru se neprovede. Nad formulářem se vypíše varování popisující vzniklou nesrovnalost v nastavení. Pro vygenerování konfiguračního souboru, musí uživatel špatné nastavení změnit a znovu stisknout tlačítko „Generovat konfigurační soubor“. Jestliže je již vše v pořádku, na stránce generátoru se místo formulářových prvků zobrazí informační text o úspěšném vygenerování a konfigurační soubor se sám nabídne ke stažení. Tato informační stránka se nachází v příloze C.2. Z této informační stránky se může uživatel vrátit zpět na nastavování hodnot, tlačítkem „Zpět na nastavení“.

Uživatel si stažený konfigurační soubor uloží do stejné složky, v jaké se nachází můj upravený model a může provést simulaci v simulátoru Ns-3, přesně dle parametrů, které si přehledně a pohodlně nastavil na webové stránce mého generátoru.

Diagram znázorňující používání výsledného webového rozhraní generátoru konfiguračního souboru se nachází v příloze B.

6 ZÁVĚR

V první kapitole bakalářské práce byly popsány sítě typu Manet, bylo zjištěno, že existuje velké množství směrovacích protokolů pro Manet sítě a bylo uvedeno, jakými způsoby se směrovací protokoly Manet sítí dělí. Některé z těchto protokolů byly stručně popsány. Pro praktickou část se bylo nutné naučit pracovat se simulátorem Ns-3, prozkoumat jeho možnosti především pro tvorbu a simulování Manet sítí a jejich protokolů. Potřebné používané abstrakce, práce se simulátorem a modely pohybu Manet sítí byly obsahem druhé kapitoly.

Třetí kapitola se již zabývala mojí praktickou částí. Byl zde popsán můj vytvořený konfigurační soubor. Všechny jeho parametry jsou zde uvedeny a přehledně zpracovány do tabulek. Konfigurační soubor byl vytvořen pro upravený model kolegy Masára, který jej souběžně vytvářel pro svojí semestrální práci. Můj vlastní výsledný konfigurační soubor obsahuje možnost volby ze tří směrovacích protokolů – AODV, DSDV, OLSR, výběr ze dvou druhů síťového provozu a navíc ze čtyř typů pohybu uzlů.

Pro pohodlné nastavování mého konfiguračního souboru, byla vytvořena webová aplikace, která konfigurační soubor dle parametrů vygeneruje a nabídne uživateli ke stažení. Webová aplikace byla vytvořena v jazyce PHP s využitím frameworku Nette. Prostředky pro vývoj webové aplikace se zabývá čtvrtá kapitola, kde byly všechny použité technologie popsány. Jsou zde popsány i všechny použité prvky uživatelského rozhraní, sloužící uživatelům pro nastavování voleb a k vygenerování souboru.

Jakým způsobem byly prvky uživatelského rozhraní a generování souboru naprogramovány, bylo popsáno v poslední páté kapitole. Přehledně je uveden kód každé naprogramované části s jeho vysvětlením. Během programování byla webová aplikace testována v pěti nejběžnějších prohlížečích a rozdíly jsou v práci taktéž uvedeny. Webová aplikace byla testována i pomocí validátorů. Poslední podkapitolou je shrnuto, jak se webová aplikace používá z pohledu uživatele. V přílohách je zobrazen doplňující diagram a snímky obrazovky webové aplikace.

Výsledná webová aplikace je velice přehledná, elegantně zpracovaná a plně funkční. Uživatel si až v 35 nastavitelných formulářových prvcích, jako jsou textová pole a roletková menu, přesně nastaví parametry své sítě Manet, kterou bude chtít po té v simulátoru Ns-3 simulovat. Po stisknutí tlačítka pro generování konfiguračního souboru se konfigurační soubor dle nastavených hodnot vygeneruje a nabídne uživateli ke stažení. Uživatel konfigurační soubor již pouze uloží do stejné složky, v které je můj upravený model Manet sítě. Po spuštění simulace v simulátoru Ns-3 se provede simulace podle uživatelského nastavení. Výsledná webová aplikace může být přínosná například při výuce, kdy student bez znalosti parametrů modelu může jednoduše vytvořit konfigurační soubory dle jeho požadavků a pozorovat rozdíly v provedených simulacích.

LITERATURA

- [1] JUBIN, John a Janet D. TORNOW. *The DARPA Packet Radio Network Protocols* [online]. Associate IEEE, leden 1987 [cit. 2012-10-17]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1457970&isnumber=31369>
- [2] Switches - What are forwarding modes and how do they work?. *Laptop, Notebook, Desktop, Server and Embedded Processor Technology - Intel* [online]. Intel Corporation, 8.9.2004, 13.6.2010 [cit. 2012-10-17]. Dostupné z: <http://www.intel.com/support/express/switches/sb/cs-014410.htm>
- [3] MOHAPATRA, Prasant a Srikanth V. KRISHNAMURTHY. *AD HOC NETWORKS: Technologies and Protocols*. New York: Springer Science Business Media, 2005, xxiii, 270 s. ISBN 03-872-2689-3.
- [4] CIPOV, Vladimir, Jan PAPAJ, Mateusz NAWROCKI, Błażej SOŁTOWSKI a Paweł LUBARSKI. *MANET Physical Layer Analysis, MANET MAC Layer analysis, MANET Routing Protocol Analysis, MANET Self Positioning Algorithms Analysis* [online]. INDECT Consortium, 23.02.2010 [cit. 2012-10-17]. D9.8 WP9. Dostupné z: http://www.indect-project.eu/files/deliverables/public/INDECT_Deliverable_D9.8_v20100223a.pdf/at_download/file
- [5] *Ns-3: ns-3 Documentation* [online]. 13.11.2012 [cit. 2012-11-20]. Dostupné z: <http://www.nsnam.org/docs/release/3.15/doxygen/index.html>
- [6] HOŠEK, Jiří. *Komunikační technologie: Směrovací protokoly Link-State* [PDF]. Brno: VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011/2012 [cit. 2012-10-21].
- [7] *Ns-3* [online]. © 2011-2012 [cit. 2012-10-30]. Dostupné z: <http://www.nsnam.org/>
- [8] *Ns-3 Manual: Release ns-3.15* [online]. 9.10.2012, 110 s. [cit. 2012-11-03]. Dostupné z: <http://www.nsnam.org/docs/release/3.15/manual/ns-3-manual.pdf>
- [9] What is WiMAX?. *4G Products, Companies & Resources - WiMax.com* [online]. WiMax.com Broadband Solutions, 2012 [cit. 2012-12-10]. Dostupné z: <http://www.wimax.com/general/what-is-wimax>
- [10] PHP just grows & grows. *Netcraft: Internet Research, Anti-Phishing and PCI Security Services* [online]. NETCRAFT, 31.1.2013 [cit. 2013-03-10]. Dostupné z: <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>

- [11] *EasyPHP: Install a local WAMP server : PHP 5 VC9, Apache 2 VC9, MySQL 5, PhpMyAdmin, Xdebug and Modules on Windows XP/Vista/Seven* [online]. EasyPHP, 2000 - 2013 [cit. 2013-03-10]. Dostupné z: <http://www.easyphp.org/>
- [12] PHP frameworky. ŠKRÁŠEK, Jan. *Programujte.com: web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu* [online]. Programujte.com, 21.2.2008 [cit. 2013-03-10]. Dostupné z: <http://programujte.com/clanek/2008022000-php-frameworky/>
- [13] *Nette Framework: Rychlý a pohodlný vývoj webových aplikací v PHP* [online]. Nette Foundation, 2008-2013 [cit. 2013-03-10]. Dostupné z: <http://nette.org/>
- [14] RAGGETT, Dave. *Raggett on HTML 4* [online]. 2. vyd. Reading, Mass.: Addison-Wesley, ©1998 [cit. 2013-05-03]. ISBN 0201178052. Dostupné z: <http://www.w3.org/People/Raggett/book4/ch02.html>
- [15] *PHP: PHP Manual - Manual* [online]. The PHP Group, © 2001-2013, 10.5.2013 [cit. 2013-05-13]. Dostupné z: <http://www.php.net/manual/en/index.php>
- [16] Regular Expression Examples. GOYVAERTS, Jan. *Regular-Expressions.info: Regex Tutorial, Examples and Reference - Regexp Patterns* [online]. © 2003-2013, 17.6.2009 [cit. 2013-05-31]. Dostupné z: <http://www.regular-expressions.info/examples.html>
- [17] *W3Schools: Online Web Tutorials* [online]. Refsnes Data, © 1999-2013 [cit. 2013-05-31]. Dostupné z: <http://www.w3schools.com/>

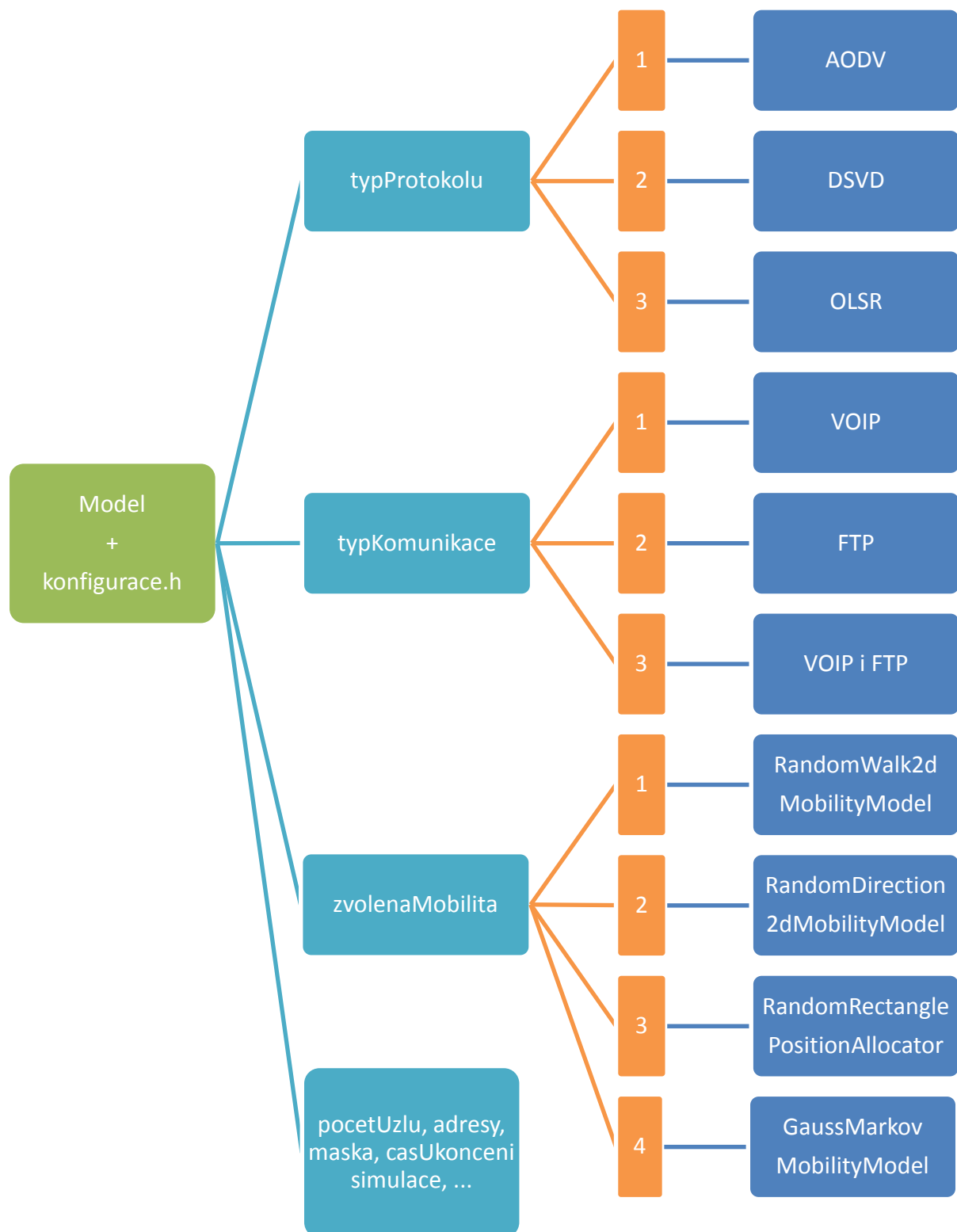
SEZNAM POUŽITÝCH ZKRATEK

AODV	Ad-hoc On-demand Distance Vector [4]
API	Application Programming Interface
CERN	Conseil Européen pour la Recherche Nucléaire
CRC	Cyclic Redundancy Check [2]
CSS	Cascading Style Sheets
DARPA	Defense Advanced Research Projects Agency [1]
DSDV	Destination Sequenced Distance Vector [4]
DSR	Dynamic Source Routing [4]
FTP	File Transfer Protocol
GPLv2	General Public License version 2
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LTE	Long Term Evolution [9]
MANET	Mobile Ad-Hoc Network [3]
NS-3	Network Simulator 3 [7]
OLSR	Optimized Link State Routing [4]
PHP	PHP: Hypertext Preprocessor
PRNET	Packet Radio Network [1]
RIP	Routing Information Protocol
SSID	Service Set Identifier
VOIP	Voice over Internet Protocol
W3C	World Wide Web Consortium
WiMAX	Worldwide Interoperability for Microwave Access [9]
ZRP	Zone Routing Protocol [4]

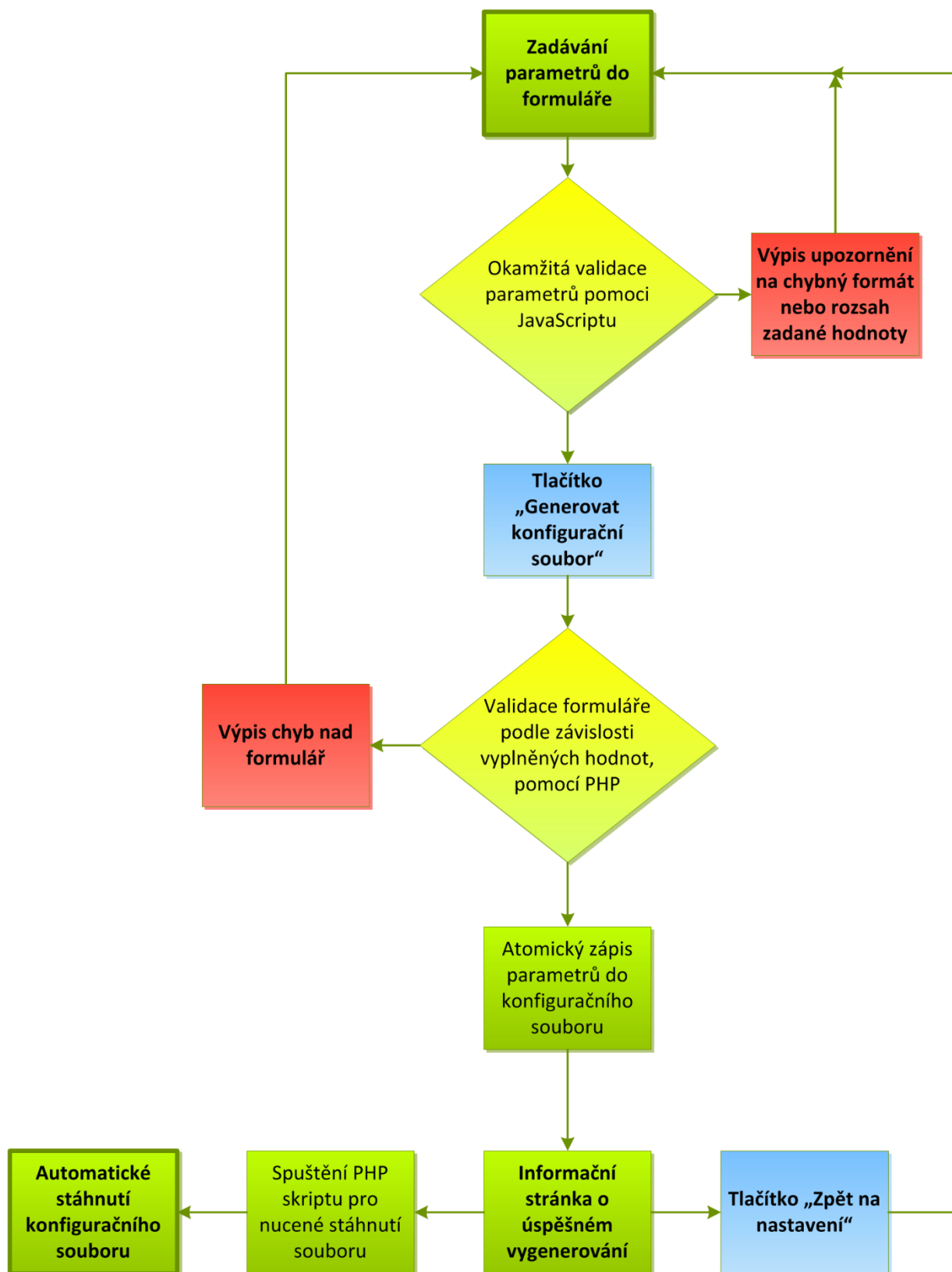
SEZNAM PŘÍLOH

A	Schéma modelu.....	52
B	Diagram funkce webové aplikace generátoru	53
C	Webová stránka generátoru	54
C.1	Konfigurační část	54
C.2	Informační text o úspěšném vygenerování	55
D	Obsah přiloženého DVD	56
E	Návod ke spuštění webu generátoru na lokálním webovém serveru	57

A SCHÉMA MODELU




B DIAGRAM FUNKCE WEBOVÉ APLIKACE GENERÁTORU



C WEBOVÁ STRÁNKA GENERÁTORU

C.1 Konfigurační část

Generátor konfiguračního souboru



Základní nastavení

Směrovací protokol:	AODV
Typ komunikace:	VOIP i FTP
Použitá mobilita uzlů:	RandomWalk2dMobilityModel
První IP adresa pro dané uzly:	10.1.1.0
Maska dané sítě:	255.255.255.0
Počet uzlů:	10
Délka simulace (s):	30
SSID:	Moje Manet sit
Název animačního souboru:	Animace Manet site
Název souboru pro statistiky toku dat:	Statistiky toku
Interval tisku směrovací tabulky do souboru (s):	2

Nastavení VOIP

První VOIP aplikace z uzlu:	0
Druhá VOIP aplikace z uzlu:	2
Komunikace bude na uzlu:	5
Čas začátku první VOIP aplikace (s):	2
Čas začátku druhé VOIP aplikace (s):	8
Čas konce první VOIP aplikace (s):	5
Čas konce druhé VOIP aplikace (s):	10
Čas začátku příjmu aplikace na klientovi (s):	1
Čas ukončení příjmu aplikace na klientovi (s):	10
Velikost jednoho paketu (B):	1024
Rychlost přenosu (Kb/s):	2048
Délka běhu komunikace VOIP (s):	1
Délka pozastavení komunikace VOIP (s):	0,01
Nastavení portu VOIP:	5060

Nastavení FTP

FTP přenos z uzlu:	2
FTP přenos na uzlu:	1
Čas začátku aplikace pro přenos FTP (s):	5
Čas konce aplikace pro přenos FTP (s):	10
Čas začátku aplikace pro příjem dat FTP (s):	5
Čas ukončení aplikace pro příjem dat FTP (s):	10
Maximální množství dat pro přenos (B):	10000
Velikost jednoho paketu (B):	128
Nastavení portu FTP:	20

Generovat konfigurační soubor

Jak používat

Dle Vašich požadavků vyplňte všechny parametry ve formuláři a stiskněte tlačítko "Generovat". Generátor vytvoří konfigurační soubor dle Vašeho zadání a nabídne Vám ho ke stažení. Výsledný konfigurační soubor zkopírujte do stejné složky, ve které se nachází můj model Manet sítě.


O stránce


Tato webová stránka slouží ke generování konfiguračního souboru sítě Manet pro simulátor Ns-3. Konfigurační soubor je určen pro použití s mým modelem. Stránku vytvořil Aleš Gilík jako součást své bakalářské práce na Fakultě elektrotechniky a komunikačních technologií VUT v Brně v roce 2013.

Vytvořil Aleš Gilík
e-mail: xgilik00@stud.feec.vutbr.cz

C.2 Informační text o úspěšném vygenerování

Generátor konfiguračního souboru



 Generování konfiguračního souboru proběhlo v pořádku. Vygenerovaný konfigurační soubor se automaticky nabídl ke stažení.

V případě potřeby si ho můžete [ručně stáhnout zde](#).

[Zpět na nastavení](#)

Jak používat

Dle Vašich požadavků vyplňte všechny parametry ve formuláři a stiskněte tlačítko "Generovat konfigurační soubor". Generátor vytvoří konfigurační soubor dle Vašeho zadání a nabídne Vám ho ke stažení. Výsledný konfigurační soubor zkopírujte do stejné složky, ve které se nachází [můj model Manet sítě](#).

O stránce

Tato webová stránka slouží ke generování konfiguračního souboru sítě Manet pro simulátor Ns-3. Konfigurační soubor je určen pro použití s mým modelem. Stránku vytvořil Aleš Gilík jako součást své bakalářské práce na Fakultě elektrotechniky a komunikačních technologií VUT v Brně v roce 2013.

Vytvořil Aleš Gilík
e-mail: xgilik00@stud.feec.vutbr.cz

D OBSAH PŘILOŽENÉHO DVD

- Bakalářská práce ve formátu PDF.
- Kompletní web generátoru konfiguračního souboru, vhodný ke spuštění na webovém serveru. Web se nachází ve složce „web_generator“.
- Instalační soubor balíčku EasyPHP verze 12.1 s prostředím PHP verze 5.4.6.

E NÁVOD KE SPUŠTĚNÍ WEBU GENERÁTORU NA LOKÁLNÍM WEBOVÉM SERVERU

1. Nainstalovat webový server a prostředí PHP, případně využít předpřipravených instalačních balíčků. Při tvorbě bakalářské práce byl použit instalační balíček EasyPHP verze 12.1 s prostředím PHP verze 5.4.6, který je přiložen na DVD.
2. Nakopírovat složku „web_generatoru“ do kořenového adresáře webového serveru. V případě balíčku EasyPHP verze 12.1 se jedná standardně o tuto složku: „C:\Program Files (x86)\EasyPHP-12.1\www“.
3. Spustit například z nabídky start nebo plochy aplikaci EasyPHP, která spustí webový server a její další součásti.
4. Spustit libovolný webový prohlížeč a do adresního řádku zadat adresu „http://localhost/web_generatoru/“, případně adresu „http://127.0.0.1/web_generatoru/“.